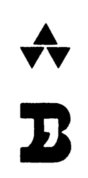


Hal Renko
Sam Edwards

Packende Spiele für Ihren Commodore 64

SPRINGER BASEL AG



Spiel Club
Band 5

Hal Renko / Sam Edwards

Packende Spiele für Ihren Commodore 64

Springer Basel AG

ISBN 978-3-7643-1648-8 ISBN 978-3-0348-6610-1 (eBook)
DOI 10.1007/978-3-0348-6610-1

CIP-Kurztitelaufnahme der Deutschen Bibliothek

Renko, Hal:
Packende Spiele für Ihren Commodore 64 / Hal Renko ; Sam Edwards. – Basel ; Boston ; Stuttgart : Birkhäuser ; London : Addison-Wesley, 1985.
(Spiel-Club ; Bd. 5)
Einheitssacht.: Crazy games for your Commodore 64 ⟨dt.⟩
ISBN 978-3-7643-1648-8

NE: Edwards, Sam:

Ursprünglich erschienen bei Birkhäuser Verlag, Basel 1985
Umschlaggestaltung: Bruckmann & Partner, Basel

Inhalt

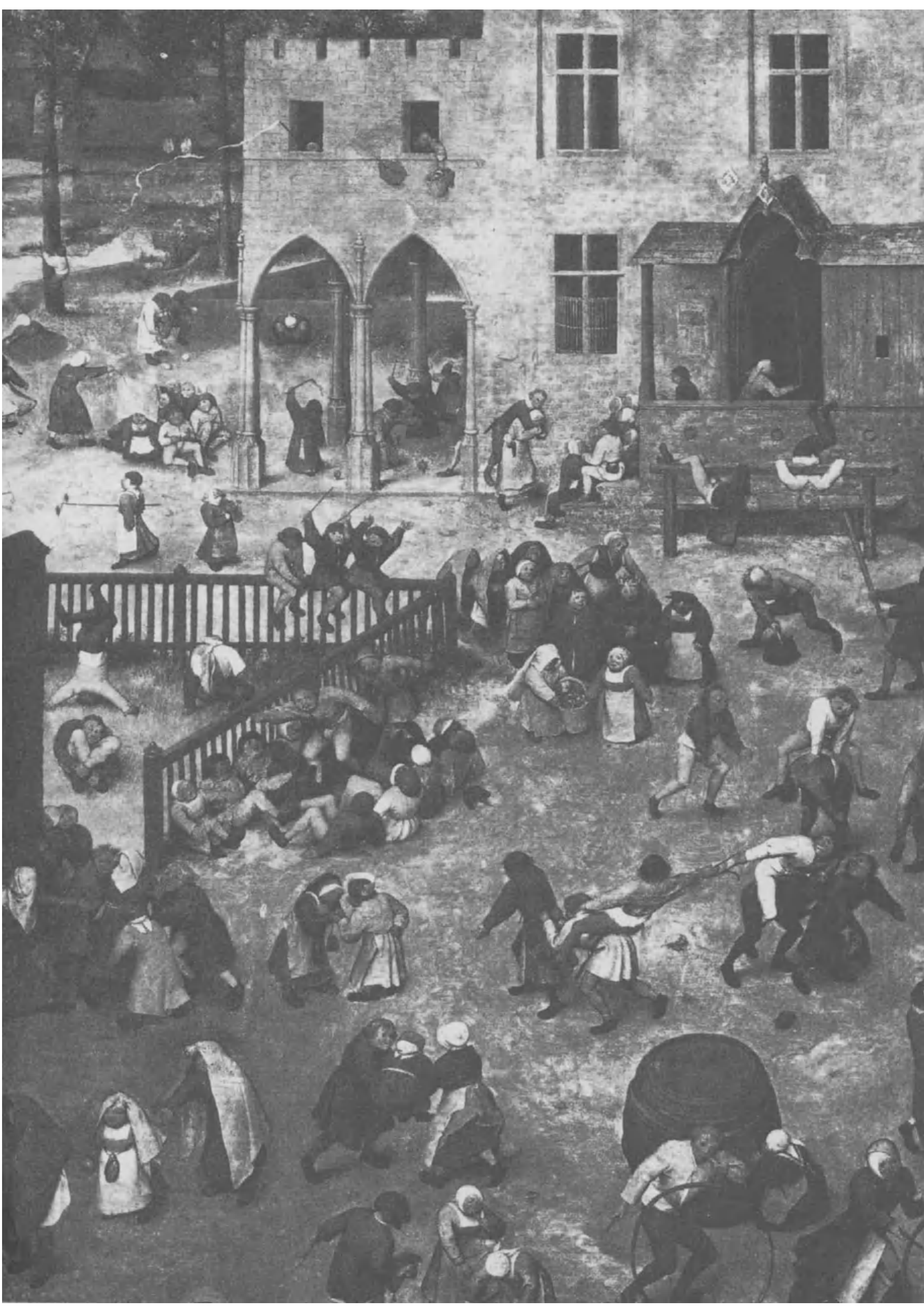
6 Danksagung
8 Einleitung
10 Galaktische Monster
14 Zombies im Sumpf
18 Tastenfeld Memory
21 Las Vegas a GoGo
28 Papagei
30 Kentucky Derby
35 Regenbogenquadrat
38 Qui Vive
47 K. Z. G.
49 Eins zu Fünf
52 Escher
56 Genie am Werk
59 Haifischjagd
63 Shakespeare Shuffle
67 Explosion
72 New York, New York
76 Schlüssel
79 Der Schwarze Kasten
82 Schatzsuche
85 I. T. – Das Abenteuer des Jahrhunderts
96 Der Wolf und die fünf Geißlein
100 Alphabahn
103 Auf dem Markt
105 Fallout
109 UFO Angriff
112 Mini Mancala
116 Stop it!
118 BAAB
120 Vokale und Konsonanten
122 Astrologie

Danksagung

Dieses Buch ist an der Technischen Hochschule Twente in den Niederlanden entstanden.

Dort begegneten wir den hier aufgeführten Spielfanatikern: Hermie Hermens, Peter de Vries, Ari-Jan Stapel, Hans Riesebos, Arend Rensink, Mark Sijbenga, Eric Groenhuis, Wim Koolhoven, Ronald Bosch, Berry Hoogstraten, Evert-Jan Ausems, Claudia Dispa, Alex Pelsmaeker und Kasper Boon.

Wir möchten ihnen allen herzlich danken. Sie haben viele Stunden mit uns verbracht und die Programme für die hier enthaltenen Spiele gestaltet und geschrieben.



Einleitung

Jeder weiß, daß Computer für alle möglichen ernsthaften Zwecke verwendet werden können.

Finanzkalkulationen, industrielle Anwendungen und Textverarbeitung, zum Beispiel. Sie können uns jedoch auch auf andere Weise helfen.

Wie schon das englische Sprichwort sagt: "All work and no play makes Jack a dull boy"; es ist sicherlich wahr, daß wir alle etwas brauchen, um die kleinen Rädchen in unserem Oberstübchen am Laufen zu halten, wenn wir uns nicht gerade mit den "wichtigen" Dingen im Leben beschäftigen. Deshalb lesen wir Bücher, pflegen Hobbies und spielen.

Brueghels wunderschönes Gemälde "Kinder im Spiel" zeigt uns, daß dies seit Jahrhunderten wahr ist, wenn nicht seit jeher.

Dieses Buch zielt darauf ab, für jedermann zwischen 6 und 96 etwas zu bieten. Es gibt Spiele, die man für sich spielt, und andere, die man mit Verwandten oder Freunden spielen kann.

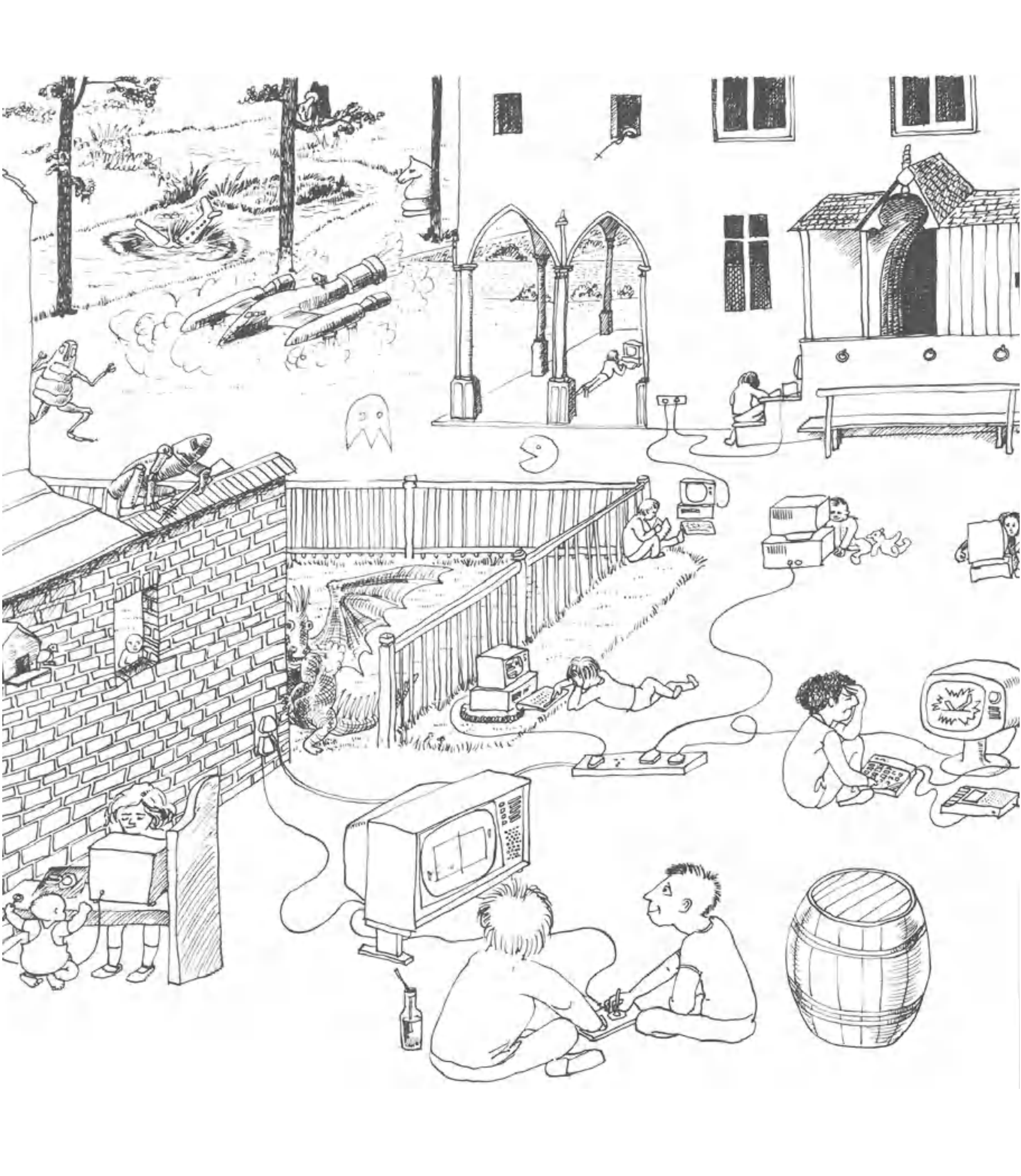
Sie werden schnelle Aktion-Spiele finden, um ihre Fähigkeit zu testen, viele sich bewegende Objekte auf dem Bildschirm zu steuern.

Puzzles und Knacknüsse, um Sie nachdenken zu lassen; Brettspiele, bei denen Sie Ihr Können gegen den Computer einsetzen; und natürlich einige wunderbare Labyrinthspiele, die vielfältige Anforderungen stellen.

Eine Anzahl Spiele wurden so ersonnen, daß sie mit vergleichsweise kurzen Programmen, die wenig Zeit zum Eintippen benötigen, interessante Ergebnisse erzielen, die der Mühe wert sind.

Wenn man sich "Kinder beim Spiel" nochmals ansieht, fällt einem etwas ein.

Falls Brueghel heute leben würde, sähe sein Gemälde ungefähr so aus . . .



Galaktische Monster

Dort ist sie, entdeckt im x-y-Gz Radar. Endlich nähern Sie sich der Milchstraße. In nur 2,56 Protosekunden sind Sie sicher in Ihrer Heimatgalaxie!

Aber es lauern Gefahren hier, und als Kommandant Ihres Schiffes wissen Sie nur zu gut, worum es geht.

Sie behalten den XR-6-Schirm genau im Auge. Noch keine Alarmbereitschaft. Die Spannung ist unerträglich ... 2,5 Protosekunden ... 2,∅ Protosekunden ... 1,5 Protosekunden ... oh nein! Dort ist er –

DER VAN ALLEN ECKSTEIN GUERTEL

Es gibt einige von diesen berüchtigten Gürteln, und in ihnen lauern die fürchterlichsten galaktischen Monster.

Keine Waffe kann Sie gegen diese rachsüchtigen Biester schützen, alles was Sie tun können, ist ihnen auszuweichen.

Wenn das Programm läuft, und Sie einen VAN ALLEN ECKSTEIN GÜRTEL sehen, dann können Sie sich durch Tastendruck bewegen:

W, für ein Feld nach oben

A, für ein Feld nach links

D, für ein Feld nach rechts

X, für ein Feld nach unten

Diese schrecklichen galaktischen Monster werden sich immer näher heranwagen.

Ihre einzige Überlebenschance ist auszuweichen! Wenn Sie einem Van Allen Eckstein Gürtel ein zweitesmal begegnen, werden die Monster noch aggressiver werden. Irgendwie scheinen sie jede Ihrer Bewegungen vorauszuahnen.

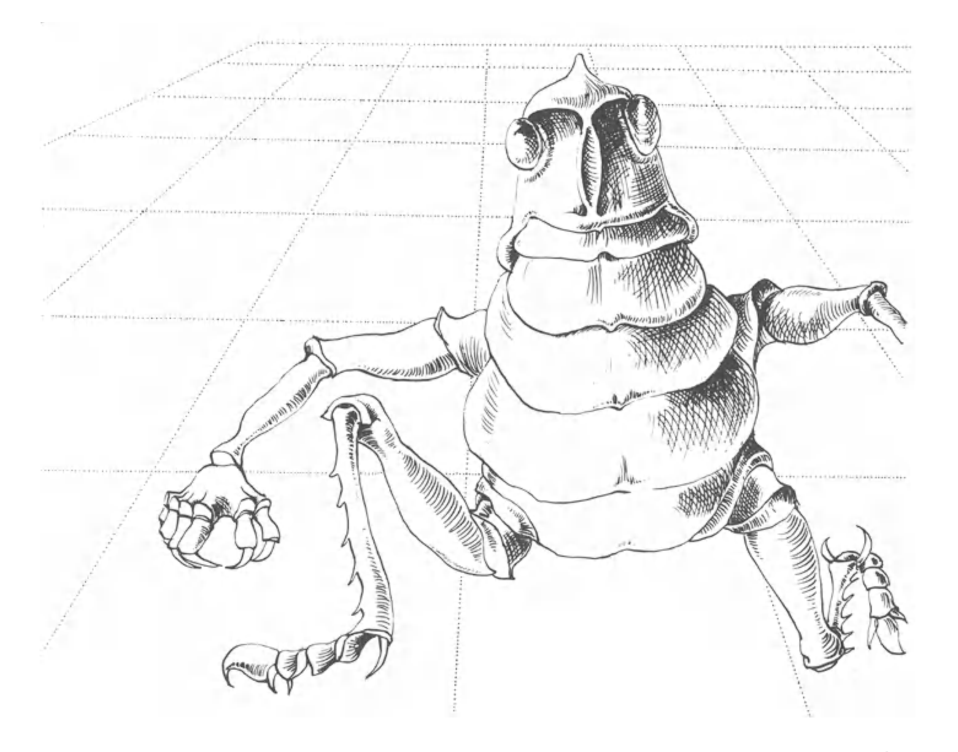
Viel Glück, Kommandant ... nur noch ein paar Protosekunden zu fliegen, und Sie sind sicher zu Hause!

```
10 rem * c-64 galaktische monster *
20 d$="[illegible]"
30 s=54272
35 for x=s to s+24:pokex,0:next
36 pokes+24,15
37 pokes,0
38 pokes+1,30
39 pokes+5,15:pokes+6,250
40 dim x(12)
50 dim y(12)
60 goto 2000
100 rem * bildschirm *
110 print"[illegible]"
120 for t=1 to 10
```

```
130 print spc(13) "✧ QQQQQQQQQQ"
140 next t
150 print spc(13) " ▒▒▒▒▒▒▒▒▒▒▒"
160 print"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒X"
170 x1=6
180 y1=2
190 for t=1 to 11
200 x(t)=t
210 y(t)=11
220 next t
230 ka=ka+.1
240 return
300 rem * bewegung des spielers *
310 get ky$:if ky$="" then 470
320 poke s+4,33
340 for m=1 to 40:next m
350 poke s+4,0
370 print "▒";tab(x1+13);left$(d$,y1);"✧
Q"
380 if ky$="d" then x1=x1+1
390 if ky$="w" then y1=y1-1
400 if ky$="x" then y1=y1+1
410 if ky$="a" then x1=x1-1
420 if x1<1 then x1=1
430 if y1<1 then y1=1
440 if x1>11 then x1=11
450 print "▒";tab(x1+13);left$(d$,y1);"▒
X"
460 print "▒";tab(x1+13);left$(d$,y1);"▒
X"
470 for t=1 to 11
480 if x1<>x(t) then 520
490 if y1<>y(t) then 520
500 gosub 1020
510 ca=-1
520 next t
530 return
600 rem * bewegungen des monsters *
610 for t=1 to 11
620 print "▒";tab(x(t)+13);left$(d$,y(t)
);"✧Q"
630 if rnd(0)<ka then 900
640 a=int(rnd(0)*4)
650 if a=0 then 690
660 if a=1 then 710
670 if a=2 then 730
680 goto 750
690 x(t)=x(t)+1
700 goto 760
710 x(t)=x(t)-1
720 goto 760
```

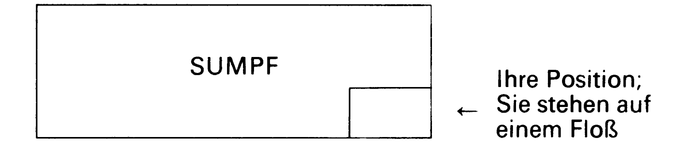
```
730 y(t)=y(t)-1
740 goto 760
750 y(t)=y(t)-1
760 if x(t)<1 then 810
770 if x(t)>11 then 830
780 if y(t)<1 then 850
790 if y(t)>11 then 870
800 goto 880
810 x(t)=x(t)+1
820 goto 960
830 x(t)=x(t)-1
840 goto 960
850 y(t)=y(t)+1
860 goto 960
870 y(t)=y(t)-1
880 print "▤";tab(x(t)+13);left$(d$,y(t)
);"▓▒"
890 goto 960
900 x2=x1-x(t)
910 y2=y1-y(t)
920 if rnd(0)<.8 then 950
930 x(t)=x(t)+sgn(x2)
940 goto 960
950 y(t)=y(t)+sgn(y2)
960 print "▤";tab(x(t)+13);left$(d$,y(t)
);"✣▒"
970 next t
980 foru=1 to 11
990 print "▤";tab(x(u)+13);left$(d$,y(u)
);"✣▒"
1000 next u
1010 return
1020 poke s+4,33
1030 for l=1 to 10
1040 for m=10 to 30+l*2
1050 poke s+1,m
1060 next m
1070 next l
1080 poke s+1,20
1090 poke s+4,0
1100 return
2000 rem * hauptprogramm *
2010 ka=ka+.1
2020 gosub 110:rem bildschirm
2030 gosub 310:rem bewegungen des spiele
rs
2040 br=br+1
2050 if not ca then 2170
2060 k=k+1:ca=0
2070 sc=sc+at-br-5*k
2080 at=at+20
```

```
2090 br=0
2100 if k<5 then 2020
2110 print"☒ende dieser runde."
2120 print"ihre punktzahl:";
2130 print sc
2140 print"fuer eine neue runde run eing
eben."
2160 end
2170 if y1=11 then 2020
2180 gosub 610
2190 goto 2030
```



Zombies im Sumpf

Vorsicht! Hier hausen Zombies, und Sie müssen den Sumpf überqueren, in dem sie leben. Dazu müssen Sie über Floße gehen. Am Anfang des Spieles sieht der Sumpf so aus:



Sie sind in der unteren rechten Ecke, und Sie müssen die schräg gegenüberliegende Ecke erreichen, um zu entkommen.

Mit Hilfe der passenden Cursortasten legen Sie ein Floß neben dasjenige, auf dem Sie momentan stehen, und durch nochmaliges Drücken steigen Sie auf das neue.

So weit, so gut, aber Vorsicht! Von Zeit zu Zeit wird eines dieser Zombies aus dem Sumpf steigen und über die Flöße rennen, die Sie verlegt haben. Wenn Sie ihm in die Quere kommen, wird es Sie fangen. Glücklicherweise können Zombies nicht besonders gut sehen, daher fallen sie oft von den Flößen herunter, zurück in den Sumpf.

Um dieses spannende Spiel erfolgreich zu spielen, müssen Sie ihre eigene Taktik entwickeln. Je länger und komplizierter Ihr Pfad ist, um so kleiner ist die Wahrscheinlichkeit, daß die Zombies Sie erwischen werden. Andererseits brauchen Sie so länger, um über den Sumpf zu kommen und geben mehr Zombies Gelegenheit zu erscheinen.

Oh nein! Hier kommen wieder ein paar dieser schrecklichen Zombies. Sie sollten besser einfach hoffen, daß sie Sie nicht erwischen.



```
10 rem * c-64 zombies im sumpf *
20 dim a(14,9)
30 for c=0 to 14
40 a(c,0)=-1:a(c,9)=-1
50 next
60 for c=0 to 9
61 a(0,c)=-1:a(14,c)=-1
63 next
65 poke 53280,2
70 a(1,0)=1:a(0,1)=1
80 def fnr(x)=int(x*rnd(0)+1)
90 em$="                    "
110 pl$="###        ###        ###"
130 mo$="***        ***        ***"
150 dr$="                *         "
170 d$="
"
180 goto 2000
200 rem * ausdrucken t$ auf x,y *
210 print left$(d$,3*(y-1)+1);tab(3*(x-1
));t$;
220 return
300 rem * initialisieren *
310 print "";
320 for li=1 to 24
330 print "
          ":rem 39 leerzeichen
340 next
350 m=0:px=13:py=8:a(13,8)=1
360 x=px:y=py
370 t$=pl$:gosub 200
380 return
400 rem * spieler bewegen *
410 rem input
420 get in$:if in$="" then 420
430 dx=(in$="")-(in$="")
440 dy=(in$="")-(in$="")
470 if dx=0 and dy=0 then 420
480 on a(px+dx,py+dy)+1 goto 500,550
490 goto 420
500 rem * floss legen *
510 x=px+dx:y=py+dy
520 t$=em$:gosub 200
530 a(x,y)=1
540 return
550 rem * bewegen *
560 x=px:y=py
570 t$=em$:gosub 200
580 px=px+dx:py=py+dy
590 if px=0 or py=0 then 630
600 x=px:y=py
```

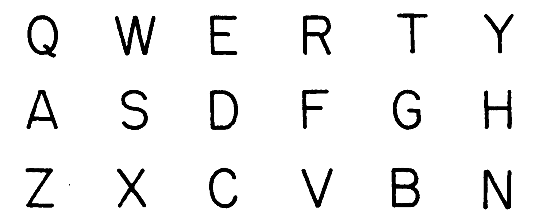
```
610 t$=pl$:gosub 200
620 return
630 rem * gewonnen *
640 print d$;"     gratuliere !";
650 for i=1 to 10:get s$:next
660 get s$:if s$="" then 660
670 run
700 rem * monster *
710 mx=13:my=8
715 x=mx:y=my
720 t$=mo$:gosub 200
730 on fnr(2) gosub 880,890
740 lx=dx:ly=dy
750 rem * neue richtung waehlen *
760 if a(mx+lx,my+ly)<1 then 800
770 if a(mx+ly,my-lx)=1 or a(mx-ly,my+lx
)=1 then 840
780 rem * richtung behalten *
790 dx=lx:dy=ly
795 goto 920
800 rem * im sumpf *
810 on fnr(4) gosub 880,890,900,910
820 if a(mx+dx,my+dy)=-1 then 810
830 goto 920
840 rem * moegliche richtung *
850 on fnr(4) gosub 880,890,900,910
860 if dx=-lx and dy=-ly or a(mx+dx,my+d
y)<1 then 780
870 goto 920
880 dx= 0:dy=-1:return
890 dx=-1:dy= 0:return
900 dx= 0:dy= 1:return
910 dx= 1:dy= 0:return
920 rem * bewegen *
930 x=mx:y=my
940 t$=em$:gosub 200
950 mx=mx+dx
960 my=my+dy
970 lx=dx:ly=dy
980 if a(mx,my)=0 then 1030
990 x=mx:y=my
1000 t$=mo$:gosub 200
1010 if mx=px and my=py then 1080
1020 goto 750
1030 rem * monster ertrunken *
1040 a(mx,my)=-1
1050 x=mx:y=my
1060 t$=dr$:gosub 200
1070 return
1080 rem * spieler erwischt *
1090 print d$;"sie sind gescheitert !";
```

```
1100 for i=1 to 10:get s$:next
1110 get s$:if s$="" then 1110
1120 run
2000 rem * hauptprogramm *
2010 gosub 300:rem initialisieren
2020 gosub 400:rem zug des spielers
2030 m=m+1
2040 if fnr(px+py)>4 or m<5 then 2020
2050 gosub 700:rem monster
2060 goto 2020
```

Tastenfeld Memory

Dieses Spiel verwendet fast das gesamte Tastenfeld des Computers, um Ihr Gedächtnis zu testen. Achtzehn Tasten verbergen je eine Zahl, in der gleichen Weise, wie eine Spielkarte mit dem Rücken nach oben ihren Wert verbirgt. Es gibt neun verschiedene Zahlen, jede versteckt unter zwei Tasten.

Die Symbole der Tasten, auf die Sie sich konzentrieren sollen, werden auf dem Bildschirm gezeigt.

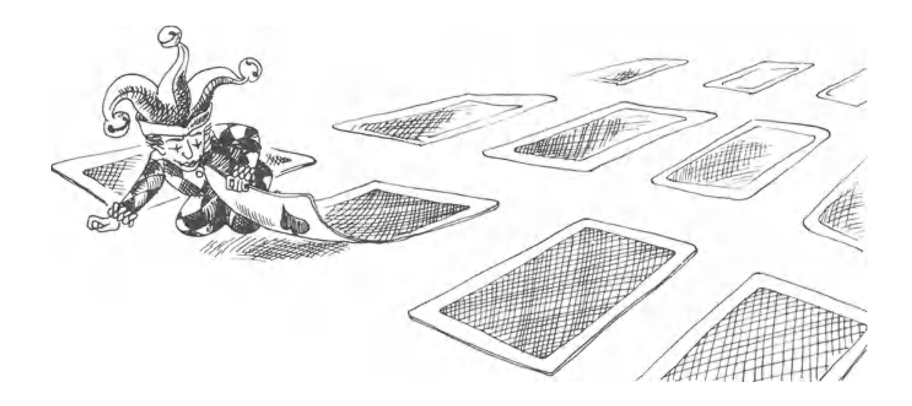


Drücken Sie eine Taste, und die Zahl, die die Taste versteckt hält, wird auf dem Bildschirm gezeigt. Drücken Sie zwei nacheinander, dann verschwinden ihre Symbole vom Schirm, wenn sie beide die gleiche Zahl verbergen.

Anfangs werden Sie natürlich raten müssen, aber wenn Sie sich die Positionen der gezeigten Zahlen merken, sollten Sie bald herausbekommen, wo die Paare liegen. Am Ende des Spiels sagt Ihnen der Computer, wie groß Ihre intellektuellen Fähigkeiten sind.

```
10 rem * c-64 tastenfeld memory *
20 d$="[illegible]"
30 r=int(rnd(0)*9+1)
40 print"[illegible]"
50 dim t$(18),w$(18),q$(2),nn(2)
60 a$="qwertyasdfghzxcvbn"
70 b$="153997231546682847"
80 goto 500
90 rem * zufaelliges erstellen des felde
s *
100 for n=1 to 18
110 t$(n)=mid$(a$,n,1)
120 di=n+r
130 if di<19 then 150
140 di=di-18
150 w$(n)=mid$(b$,di,1)
160 next n
170 return
180 rem * bildschirm *
190 for k=1to 3
200 for j=1 to 6
```

```
210 print"■"
220 print right$(d$,3*k);tab(5*j);
230 print t$((k-1)*6+j)
240 nextj,k
250 return
260 rem * eingaben *
270 print
275 q$(2)=""
280 for kk=1 to 2
290 get ky$
300 if ky$="" then 290
305 if ky$=q$(1) then 290
310 q$(kk)=ky$
320 tu=tu+1
330 rem * karte umdrehen *
335 nn(kk)=0
340 for n=1 to 18
350 k=int((n-.5)/6)+1
360 j=n-(k-1)*6
370 if t$(n)<>q$(kk) then 410
380 print"■";right$(d$,3*k+1);tab(5*j);
390 print w$(n);
400 nn(kk)=n
410 next n
415 if nn(kk)=0 then 290
420 next kk
430 return
440 rem * karten vergleichen *
450 if w$(nn(1))<>w$(nn(2)) then return
460 te=te+1
470 t$(nn(1))=" "
480 t$(nn(2))=" "
490 return
```



```
500 rem * hauptprogramm *
510 gosub 100:rem zufallsfeld
520 gosub 190:rem bildschirm
530 gosub 270:rem eingaben
540 gosub 450:rem karten vergleichen
550 if te<9 then 520
560 print "[illegible]zahl der umgedrehten karten
:";tu
570 end
```

Las Vegas a GoGo

Haben Sie schon mal beobachtet, wie jemand Münze nach Münze in einen einarmigen Banditen geworfen hat, und sich gewundert, was da so faszinierendes dran ist? Sie werden es schnell herausfinden, weil dieses Programm Ihren Computer in eine dieser sagenhaften Las Vegas Slotmaschinen verwandelt.

Alle Spielanweisungen, die Sie brauchen, werden auf dem Bildschirm erscheinen.

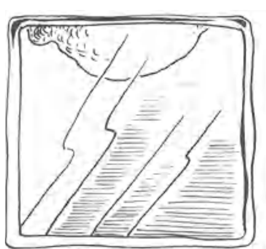
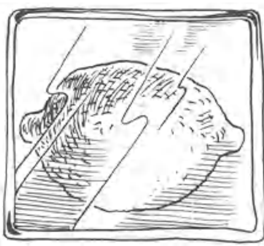
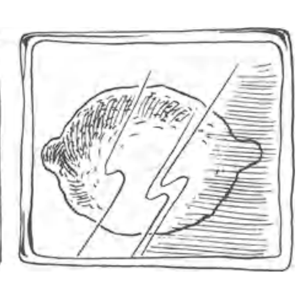
An bestimmten Stellen bekommen Sie eine Liste mit Wahlmöglichkeiten, zum Beispiel

EINWERFEN, HALTEN, SPIELEN, BEENDEN

Geben Sie Ihre Wahl ein, indem Sie den Anfangsbuchstaben der Möglichkeit, die Sie wollen, eingeben. Beispielsweise hält S die Maschine am Laufen.

Die Räder sind mit 1, 2 und 3 numeriert. Wenn Sie ein Rad oder mehrere Räder halten wollen, tippen Sie die entsprechende Zahl oder Zahlen ein, nachdem Sie H eingegeben haben.

Lichter werden blitzen und Musik wird ertönen, wenn die Räder sausen. Haben Sie dieses Mal gewonnen? Macht nichts, Sie machen sicher den Hauptgewinn . . . früher oder später!

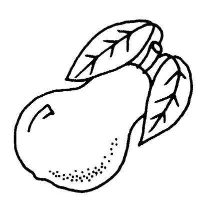
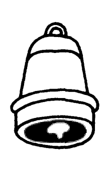
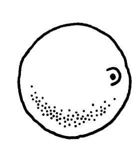




```
2 rem * c-64 las vegas a gogo *
5 poke 53281,7
10 sc=1024:cl=55296:cr=54272
20 def fnr(x)=int(rnd(1)*x+1):x=rnd(-ti)
100 dimwf$(13),r(2,13),ja(13):ja(5)=-1:j
a(6)=-1:ja(7)=-1
110 for v=1 to 13:read wf$(v),r(1,v),r(2
,v):next
120 dataa,,,k,,,q,,,j,,,*,8,40,"Q",8,40,
$,7,30
130 data"A",5,25,"S",3,20,"Z",3,20,"X",2
,10,#,2,10,%,,
200 dim jp(4),jf(4),jc(4)
210 for jo=1 to 4:read jp(jo),jf(jo):nex
t
```

```
220 data 293,1,294,11,296,17,297,10
300 dim wp(3),v(4),hv(3):wp(1)=211:wp(2)
=213:wp(3)=215
310 dim hp(3),hf$(3):hp(1)=453:hp(2)=455
:hp(3)=457
320 ho$="hhh"
400 dim gp(4),gv(4):gp(1)=32:gp(2)=76:gp
(3)=112:gp(4)=69
410 br$="                "
500 nj=4:nh=0:ng=0:gw=0:ni=0:nt=-1
510 pl$=""
515 rem 39 leerzeichen
520 h$=left$(pl$,12)+""
530 for l=cr to cr+24: poke l,0: next
540 poke cr+6,134: poke cr+5,16
550 hf=88:lf=115:dr=40
560 poke cr+24,15
570 goto 3000
599 rem inkrement zum geld addieren
600 for ad=sgn(in) to in step sgn(in)
610 print"total:"mo+ad"   "
620 if in>0 then gosub 2700
630 if in<0 then for i=0 to 2:hf=38:lf=1
26:gosub 2700
640 next ad
650 hf=88:lf=115
660 mo=mo+in:return
699 rem blink s1$/s2$ and get in$
700 print "" s1$:for de=1 to 250:nex
t
710 get in$:if in$<>"" then print ""
s2$: return
720 print "" s2$:for de=1 to 250:next
730 get in$:if in$<>"" then return
740 goto 700
799 rem * doppelte beseitigen
800 do=0:print"              "
810 for jo=1 to 4
820 pokesc+jp(jo)+3,jf(jo):pokecl+jp(jo)
+3,0:jc(jo)=0:next
830 nj=4:return
899 rem * jackpot *
900 hf=64:lf=188
910 for i=1 to 4:print"
 ";
920 for c=1 to jc
930 print "j";:gosub 2700
940 next c,i
950 hf=88: lf=115
960 return
```

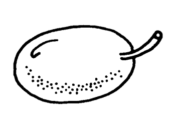
```
1000 gosub 2700:print"████total: 0"
1010 print"█████"tab(15)"██       ██ "
1020 for i=1 to 4:printtab(15)"██
█ ":next
1030 printtab(15)"██         "
1040 printtab(15)"█ █      █ "
1050 for i=1 to 6:printtab(16)"██      ":
next
1060 printtab(15)"██        █"
1070 printtab(14)"██          █"
1100 print"████████████ %  =  1"
1105 print"███████"
1110 for v=12 to 5 step-1
1120 printtab(3)"█-"wf$(v)wf$(v)"-= "r(1
,v)
1125 next
1130 print"██████"
1140 for v=12 to 5 step-1
1150 print tab(28)"█"wf$(v)wf$(v)wf$(v)"
 ="r(2,v);
1160 if ja(v) then print"██+j";
1170 print:next:return
1199 rem * variable anpassen *
1200 nt=nt+1:if wi>0 then hb=-1:wi=0
1210 for ho=1 to 3:hf$(ho)=" ":pokesc+hp
(ho)+3,160:next
1220 if do then 1270
1230 for jo=1 to 4:if jc(jo)=0 then 1260
1240 jc(jo)=jc(jo)-1:if jc(jo)>0 then 12
60
1250 nj=nj+1:pokesc+jp(jo),jf(jo):poke c
l+jp(jo),0
1260 next jo:goto 1280
1270 dc=dc-1:if dc<=0 then gosub 800
1280 print"█████" spc(11) "
    "
1290 if ja then jc=jc-1:print"████" spc(
jc) " ":if jc<=0 then ja=0
1300 if mo>mm then mm=mo
1310 return
1399 rem * nicht genug geld *
1400 hb=-1:gosub 800
1410 ja=0:print"████                 "
1420 print pl$ tab(10) "███einwerfen ode
r beenden";
1430 s1$="███einwerfen (e)":s2$="████"+br
$:gosub 700
1440 if in$="s" then 1430
1450 return
1499 rem * moegliches halten *
```

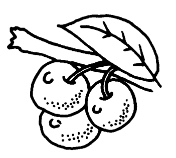
```
1500 print pl$ tab(5)"einwerfen,halten,s
pielen,beenden";
1510 s1$=h$+ho$:s2$=h$+hf$(1)+"▮▮▯"+hf$(2
)+"▮▮▯"+hf$(3)
1520 gosub 700:ho=val(in$):if ho=0 or ho
>3 then return
1530 nh=nh+1:if hf$(ho)=" " then hf$(ho)
="▩h":goto1510
1540 hf$(ho)=" "
1550 goto 1510
1599 rem * kein halten *
1600 print pl$ tab(5) "einwerfen, spiele
n oder beenden";
1610 s1$="▩▮▮▮▮???":s2$="▩▮▮▮▮   ":gosub 7
00
1620 return
1699 rem * was tun mit gewinnen? *
1700 print pl$;:if hb then 1720
1710 print tab(8) "halten, ";
1720 print tab(10) "risiko,auszahlen";
1730 goto1610
1799 rem * raeder drehen *
1800 for wd=1 to 3:if hf$(wd)<>" " then
1830
1810 fi=fnr(100):if fi<5 then v(wd)=fi-8
*(jc(fi)>0):goto 1830
1820 v(wd)=5-(fi>7)-(fi>10)-(fi>13)-(fi>
23)-(fi>36)-(fi>49)-(fi>68)-(fi>87)
1830 for de=2 to 500:next:hf=40:lf=200:g
osub 2700
```

```
1850 printh$ "▩▩▩▩" spc(wp(wd)-wp(1)) wf
$(v(wd))
1860 hf=88:lf=115
1870 next wd:return
1899 rem * auf joker achten *
1900 j=0:for wd=1 to 3:v=v(wd)
1910 if v>4 then fv=v:goto 1950
1920 jw=wd:j=j+1
1930 if jc(v)>0 then 1950
1940 print"▩▩▩" spc(jp(v)-80) "▮ ":jc(v)
=20:nj=nj-1
```

```
1950 next wd:do=(nj=0):if do then dc=15:
print"▓▓▓doppel!"
1960 return
1999 rem * gewinne berechnen *
2000 hv=13:on j goto 2030,2010,2070
2010 if not(ja and fv>10) then hv=fv
2020 goto 2070
2030 v(0)=v(3):v(4)=v(1):if ja and v(jw-
1)<>v(jw+1) then 2070
2040 v(0)=15:v(4)=15
2050 if v(jw+1)>v(jw-1) then hv=v(jw-1):
goto 2070
2060 hv=v(jw+1)
2070 for wd=1 to 3:hv(wd)=v(wd):if v(wd)
<5 then hv(wd)=hv
2080 next wd:if hv(1)<>hv(2) or hv(2)<>h
v(3) then 2100
2090 if ja(hv(1)) then ja=-1:jc=15:gosub
 900
2099 rem * gewinne berechnen *
2100 for wd=1 to 3:if hv(wd)=13 then wi=
wi+1
2110 next wd:if ja then wi=10*wi
2120 ns=-(hv(1)=hv(2))-(hv(2)=hv(3)):wi=
wi+r(ns,hv(2))
2130 if do then wi=2*wi
2140 return
2199 rem * risikoschleife *
2200 dt=1:gv(1)=2*wi:gv(2)=0:gv(3)=int(3
*wi/2):gv(4)=int(wi/2)
```





```
2210 ng=ng+1:print pl$ tab(8) "      stop
     ";
2220 forli=1to3:print"▓"spc(40*li-12) "▓
▓            ":next:rem 12 leerzeichen
2230 r=r+1:if r>4 then r=1
2240 print"▓▓" spc(gp(r)) gv(r):hf=16*r:
lf=47*r:gosub 2700:print"▓▓▓▓  "
2250 if in$<>"s" then get in$:goto 2270
2260 dt=(1+rnd(1))*dt:for de=1 to dt:nex
t:if dt>500 then 2280
2270 print"▓▓" spc(gp(r)+1) "   ":goto 2
230
```

```
2280 hf=88:lf=115:for li=1 to 3:print"▒"
 spc(40*li-12) "                ":next
2290 gw=gw+gv(r)-wi:wi=gv(r):return
2399 rem * spielende *
2400 print"▒":hf=19:lf=63:gosub 2700
2410 print"eingeworfen:";:x=ni:gosub 250
0
2420 print"ausgezahlt:";:x=mo/4:gosub 25
00
2430 print"maximum in einer runde:";:x=m
m/4:gosub 2500
2440 print"spielgewinn:";:x=gw/4:gosub 2
500
2450 print"gehalten:   ";nh
2460 print"spiele   :   ";ng
2470 print"runden   :   ";nt
2480 return
2500 if x=0 then x$=br$+".000":goto 2520
2510 x$=br$+str$(int(x*100)/100+.001)
2520 print tab(18) "$" mid$(x$,len(x$)-7
,7):print
2530 return
2600 print"▒▒▒▒▒▒▒▒▒▒" spc(23);:gosub 266
0
2610 for i=0 to 3:print"▒▒ ▒▒▒▒ ";:gosub
2660:next
2620 for wd=1 to 3:if hf$(wd)=" " then p
oke sc+wp(wd),32
2630 next wd
2640 for i=0 to 3:print"▒▒▒ ▒▒▒▒ ";:gosub
2660:next
2650 return
2660 for de=0 to 80:next
2670 return
2700 rem * sound *
2720 poke cr+1,hf:poke cr,lf
2735 poke cr+4,17
2740 for t=1 to dr:next
2745 poke cr+4,16
2750 return
2999 rem * hauptrogramm *
3000 gosub 1000:rem bildschirm
3010 gosub 1200:rem anpassen
3020 on -2*hb-(mo>1)+1 gosub 1400,1500,1
400,1600
3030 ca=-(in$="e")-2*(in$="s")-3*(in$="b
")
3040 on ca goto 3100,3200,4000
3050 goto 3020
3100 ni=ni+1:in=4:gosub 600
3110 if mo>mm then mm=mo
```

```
3120 goto 3020
3200 in=-2:gosub 600
3210 gosub2600:gosub 1800:rem raeder dre
hen
3220 if do then 3240
3230 gosub 1900:rem jokers
3240 gosub 2000:rem gewinne
3250 if hf$(1)<>" " or hf$(2)<>" " or hf
$(3)<>" " then 3280
3260 hb=0:lw=wi:if wi>0 then 3300
3270 goto3010
3280 hb=-1:if wi>lw then 3300
3290 print"▒▒▒▒▒▒"spc(11)"sie verloren "
:hf=21:lf=154:gosub 2700:fori=0to2000
3295 next:hf=88:lf=115:goto 3010
3300 print"▒▒▒▒▒▒" spc(11) "sie gewannen
 "wi"▒  "
3310 for du=1 to wi:hf=108:lf=223:gosub
2700:for i=0 to 50:next:hf=88:lf=115
3320 next du
3330 if mo<2 then hb=-1
3340 if wi>499 then 3400
3350 gosub 1700:rem anweisung holen
3360 ca=-(in$="h" and not hb)-2*(in$="r"
)-3*(in$="a")
3370 on ca goto 3600,3500,3400
3380 goto 3340
3400 in=wi:gosub 600:goto3010
3500 hb=-1:gosub 2200:rem spielen
3510 if wi>0 then 3300
3520 goto 3290
3600 wi=0:goto 3010
4000 gosub2400:rem ende
4005 poke 54296,0
4010 end
```

Papagei

Papagei ist ein sehr einfaches Reaktionsspiel. Spielen Sie es an einem regnerischen Tag mit einem Freund (besonders, wenn Ihr Freund ein paar Gläser Bier getrunken hat!).

Am Anfang des Spiels fragt Ihr Computer Sie unverschämt:

BIST DU SOWEIT, PAPAGEI?

Ja, er nennt Sie einen Papagei! Drücken Sie irgendeine Taste, und der Computer gibt sofort einen Buchstaben aus.

Sie müssen den gleichen Buchstaben so schnell wie möglich eingeben. Und weil Sie jeden Buchstaben nachtippen, sind Sie wirklich ein Papagei! Das Spiel wird wiederholt für etwa 1Ø Sekunden, und dann zeigt der Computer Ihre Punktzahl.

Wie gut sind Sie als Papagei?

Machen Sie einen Wettstreit mit Ihren Freunden und versuchen Sie den Titel eines "Super-Duper Papageis" zu gewinnen. (Mein Rekord ist 19 – Ich unterrichte Papageien!)



```
10 rem * c-64 papagei *
20 print "▣▣           " spc(10)"▣Q   "
30 print "          " spc(10)"▣     "
40 print "          " spc(10)"▮▣    "
50 print "           " spc(10) "▣    "
60 print "            " spc(10) "▣  "
70 print "     ▣" spc(12) "▣papagei"
75 print
80 a=65:b=26
90 c=1
100 print "          bist du soweit, papa
gei?"
110 get a$:if a$="" then 110
120 print
130 t=ti
140 x$=chr$(a+b*rnd(1))
145 print x$;
150 get a$:if a$<>x$ then 150
160 if ti-t>600 then 180
170 c=c+1:goto 140
180 print
190 print "punktzahl:";c
200 print "zeit:";int((ti-t)/.6)/100;"se
kunden"
```

Kentucky Derby

Meine Damen und Herren, sie sind gestartet. Red Arrow hatte einen guten Start, Blondish Boy macht seinem Jockey Schwierigkeiten . . . Und dort ist die berühmte Spanish Lady, auf der Innenbahn. Dies ist wirklich ein erstklassiges, spannendes Rennen und wir haben gerade erst angefangen! Sie kommen eben aus der ersten Kurve . . . Mickey Finn hat die Führung übernommen, hart bedrängt von Speedy Gonzales . . .

Das Kentucky Derby ist ein Pferderennspiel. Es gibt in unserem Kentucky Derby nicht so viele Pferde wie im wirklichen Rennen – es sind genau drei – aber sie müssen 2 lange Runden durchlaufen (mit Computerpferderennen ist alles möglich!).

Am Anfang des Spieles werden die Quoten für jedes Pferd bekanntgegeben.

Nun können Sie Ihre Wetten plazieren.

Solange Sie Geld haben, können Sie immer wieder wetten – diese Pferde werden nie müde.

Nach dem RUN-Befehl sehen Sie:

SETZEN SIE JETZT DIE QUOTEN:

PFERD 1: 1Ø ZU 1

PFERD 2: 4 ZU 1

PFERD 3: 2Ø ZU 1

(Die Quoten sind für jedes Rennen anders).

SIE HABEN NOCH . . . DM

IHR EINSATZ AUF PFERD 1 IST . . .

Wenn Sie kein Geld mehr haben, sagt Ihnen Ihr Computer frech:

SOVIEL KOENNEN SIE NICHT AUSGEBEN

Und wenn Sie etwas besser dran sind:

SIE HABEN NOCH . . . DM

Gleich nachdem Sie alle Ihre Wetten eingegeben haben, erscheint auf dem Bildschirm die Rennbahn mit den Pferden am Start.

Die Quoten und die Wetten werden im unteren Teil des Bildschirmes angezeigt, und dann sehen Sie:

START????

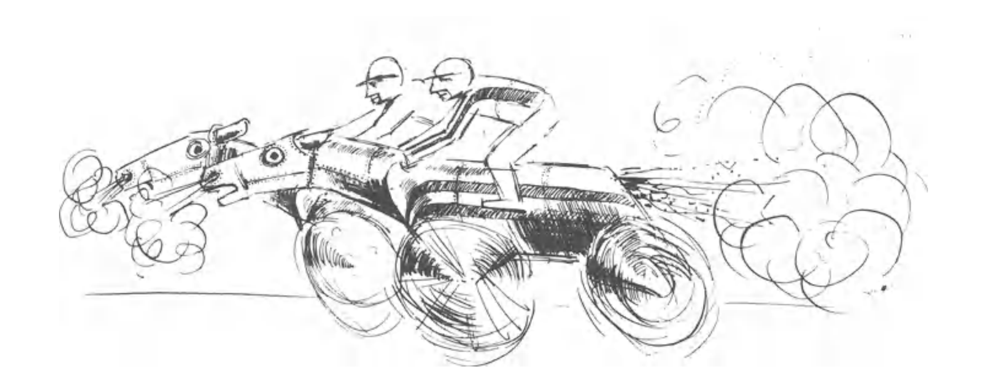
Drücken Sie RETURN, und es geht los!

```
10 rem * c-64 kentucky derby *
20 d$=""
30 gosub1400
200 rem * hauptprogramm *
220 print "";
230 gosub700
250 gosub1200
260 print""d$"                    "
270 gosub2000
280 goto400
330 print""d$"pferd "wi" hat gewonne
n !!!!!!!"
331 forh=1to3000:next
335 cr=cr+i(wi)*p(wi)
336 restore
337 foru=0to5:pokev+u,0:next
338 y1=60:y2=100:y3=140
339 x1=0:x2=0:x3=0
340 if cr>0 then 230
350 print"sie sind ruiniert !"
360 print"ich hoffe, sie haben etwas daz
ugelernt."
380 end
400 rem * einen schritt bewegen *
405 t%=rnd(ti)*3+1
410 a1=3:a2=3:a3=3
415 ift%=1thena1=5
420 ift%=2thena2=5
425 ift%=3thena3=5
430 x1=x1+a1:x2=x2+a2:x3=x3+a3
435 ifx1>252thenx1=0:y1=60:c1=c1-1
440 ifx2>252thenx2=0:y2=100:c2=c2-1
445 ifx3>252thenx3=0:y3=140:c3=c3-1
450 ifx1>110 and x1<120 then y1=y1-02:u1
=14:x1=x1-a1+2:pokev,x1:x1=x1+2
455 ifx2>110 and x2<120 then y2=y2-02:u2
=14:x2=x2-a2+2:pokev+2,x2:x2=x2+2
460 ifx3>110 and x3<120 then y3=y3-02:u3
=14:x3=x3-a3+2:pokev+4,x3:x3=x3+2
465 ifx1>160 and x1<170 then y1=y1+02:u1
=14:x1=x1-a1+2:pokev,x1:x1=x1+2
470 ifx2>160 and x2<170 then y2=y2+02:u2
=14:x2=x2-a2+2:pokev+2,x2:x2=x2+2
475 ifx3>160 and x3<170 then y3=y3+02:u3
=14:x3=x3-a3+2:pokev+4,x3:x3=x3+2
480 ifx1>=120 and x1<=160 then u1=14:x1=
x1-a1+2:pokev,x1:x1=x1+2
485 ifx2>=120 and x2<=160 then u2=14:x2=
x2-a2+2:pokev+2,x2:x2=x2+2
490 ifx3>=120 and x3<=160 then u3=14:x3=
x3-a3+2:pokev+4,x3:x3=x3+2
```

```
500 ifu1=14then u1=13:goto510
505 u1=14
510 ifu2=14then u2=13:goto520
515 u2=14
520 ifu3=14then u3=13:goto530
525 u3=14
530 poke2040,u1:poke2041,u2:poke2042,u3
540 pokev,x1:pokev+2,x2:pokev+4,x3
550 pokev+1,y1:pokev+3,y2:pokev+5,y3
570 ifc1<0thenwi=1:goto330
580 ifc2<0thenwi=2:goto330
590 ifc3<0thenwi=3:goto330
600 goto400
700 rem * pferde initialisieren *
720 wi=0
730 fork=1to3
740 x(k)=20
750 c1=1:c2=1:c3=1
760 p(k)=(int(rnd(0)*5)+1)/5
770 v(k)=p(k)*dv+vn
780 t(k)=5*(k-1)+3
800 next k
900 rem * wetten plazieren *
910 pt=0
920 fori=1to3
930 p(i)=(v(i)-vn)/dv
940 pt=pt+p(i)
950 nexti
960 fork=1to3
970 p(k)=int(pt/p(k))
980 nextk
990 print"▒";
1000 print"▒***********************"
1010 print"   setzen sie jetzt
1020 print"***********************"
1030 print
1040 print"die quoten:"
1050 print "sind:"
1060 fori=1to3
1070 print"pferd";i;": ";p(i);"zu 1"
1080 nexti
1090 print
1100 print"sie haben noch";cr;"▒ dm  "
1110 fori=1to3
1120 print"ihr einsatz auf";
1125 print"pferd";i;"ist:"
1130 inputi(i)
1140 re=cr-i(i)
1150 ifre<0 thenprint"soviel koennen sie
 nicht ausgeben":goto1120
```

```
1160 cr=cr-i(i):print"sie hab
en noch";cr;" dm  "
1170 nexti
1180 return
1200 rem * bildschirm *
1210 print"";
1220 print""
1230 print""
1240 print""
1245 print:print
1250 print"CCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCC"
1260 print"quoten:  ";p(1);tab(12);p(2);t
ab(18);p(3)
1270 print"wetten:   ";i(1);tab(12);i(2);
tab(18);i(3)
1280 print left$(d$,t(1)-3);"CCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCC"
1290 fork=1to3
1325 print left$(d$,t(k)+3);"
     NM                          "
1330 print left$(d$,t(k)+4);"CCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCC"
1340 nextk
1341 print""
1342 for r=1to16:printtab(27);"
     "
1343 nextr
1344 printtab(27);"   finish    "
1350 input"start????";a$
1370 return
1400 rem * intialisieren *
1420 vn=2:dv=1:vu=2
1430 y1=60:y2=100:y3=140
1440 cr=100:u1=14:u2=14:u3=14
1450 x1=0:x2=0:x3=0
1460 return
2000 forn=0to62:readq:poke832+n,q:next
2005 forn=0to62:readq:poke896+n,q:next
2010 data 0,0,0,0,0,0,0,0,0,0,0,24
2020 data 0,0,52,0,0,126,0,0,246,3,255,2
24
2030 data 7,255,192,15,255,128,31,255,12
8,38,1,128
2040 data 76,0,192,24,0,96,48,0,48,0,0,0
2050 data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0
2060 data 0,0,0,0,0,0,0,0,0,0,0,24
2070 data 0,0,52,0,0,126,0,0,246,3,255,2
24
```

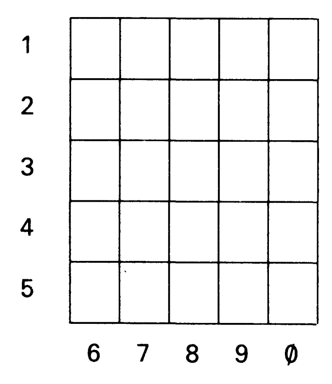
```
2080 data 7,255,192,15,255,128,31,255,12
8,38,1,128
2090 data 67,3,0,3,134,0,0,204,0
2095 data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0
2100 v=53248:pokev+21,7:form=2040to2042:
pokem,13:next:pokev+33,0
2105 pokev+23,7:pokev+29,7
2140 pokev+40,13
2150 return
```



Regenbogenquadrat

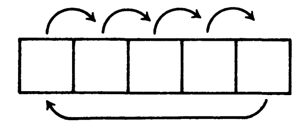
Ein zweidimensionaler Würfel? Natürlich unmöglich, aber dieses Spiel, das auf dem Computerbildschirm gespielt wird, ähnelt in vielen Aspekten Rubiks Zauberwürfel.

Sie werden 25 farbige, zufällig verteilte Felder auf einem 5 mal 5 Brett sehen. Die Reihen der Quadrate auf dem Brett sind folgendermaßen numeriert:

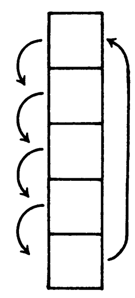


Das heißt, die waagrechten Reihen sind von 1 bis 5 durchnumeriert und die senkrechten Spalten von 6 bis 9 und dann Ø.

Wenn Sie die Nummer einer Reihe eingeben, werden die Felder in derjenigen Reihe um eine Position bewegt. In einer waagrechten Reihe wandern die Felder nach rechts, in einer vertikalen nach unten. Ein Feld, das am Ende einer Reihe verschwindet, taucht am anderen Ende wieder auf, etwa so



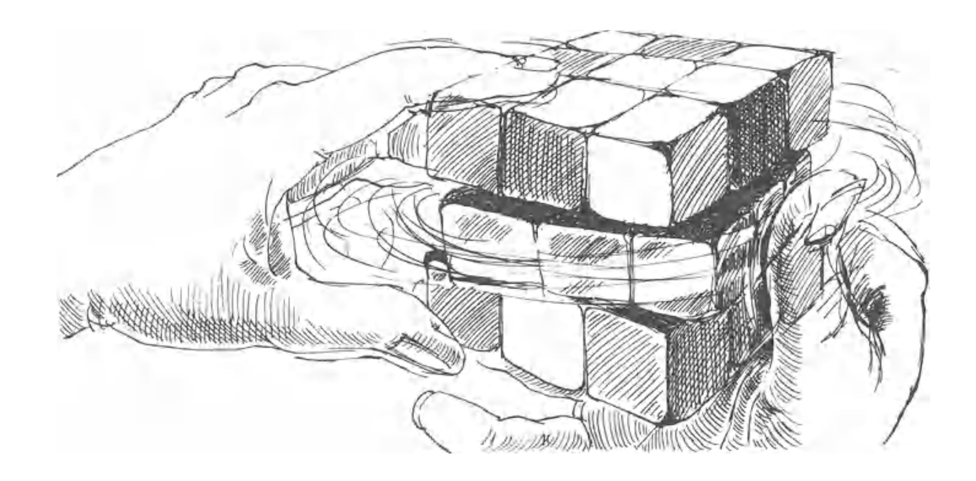
in einer waagrechten Reihe, oder so



in einer vertikalen.

Der Sinn davon? Sie müssen die Felder zu fünf waagrechten Streifen von je einer Farbe ordnen. Klingt einfach . . . aber ist es das auch?

```
10 rem * c-64 regenbogenquadrat *
20 print "▒":poke 36879,59
30 dim a%(5,5):for i=0 to 4:read a%:for
j=0 to 4:a%(i,j)=a%:next j,i
40 data28,5,31,158,30
50 :rem rot,weiss,blau,gelb,gruen
60 goto 510
100 rem * bildschirm *
110 print "▒▒▒▒▒▒▒▒▒▒▒▒▒U-6----7----8----9----0-I
"
120 for i=5 to 1 step -1:print spc(10) "
]";spc(15);"]"
130 a$=right$(str$(i),1):print spc(10) a
$;spc(15);a$
140 print spc(10)"]";spc(15);"]"
150 next:print "▒▒▒▒▒▒▒▒▒▒▒▒J-6----7----8----9----
0-K"
160 print "▒";:for i=0 to 4:for j=1 to 3
:print:print spc(10) "▒▒";:for k=0 to 4
170 print chr$(a%(i,k));"   ";:next k,j,
i
180 return
200 rem * drehen *
210 for i=5 to 1 step -1:a%(a,i)=a%(a,i-
1):next
220 a%(a,0)=a%(a,5):return
230 for i=5 to 1 step -1:a%(i,a)=a%(i-1,
a):next
```



```
240 a%(0,a)=a%(5,a):return
500 rem * hauptprogramm *
510 gosub 110:for w=0 to 20:a=int(5*rnd(
1)):on 1+2*rnd(1) gosub 210,230:gosub 16
0:next w
520 get a$:if a$="q" then sys 65234
530 a=val(a$):if a$="0" then a=10
540 if a=0 then 550
550 if a>5 then a=a-6:gosub 230:goto 570
560 a=5-a:gosub 210
570 gosub 160:goto 520
```

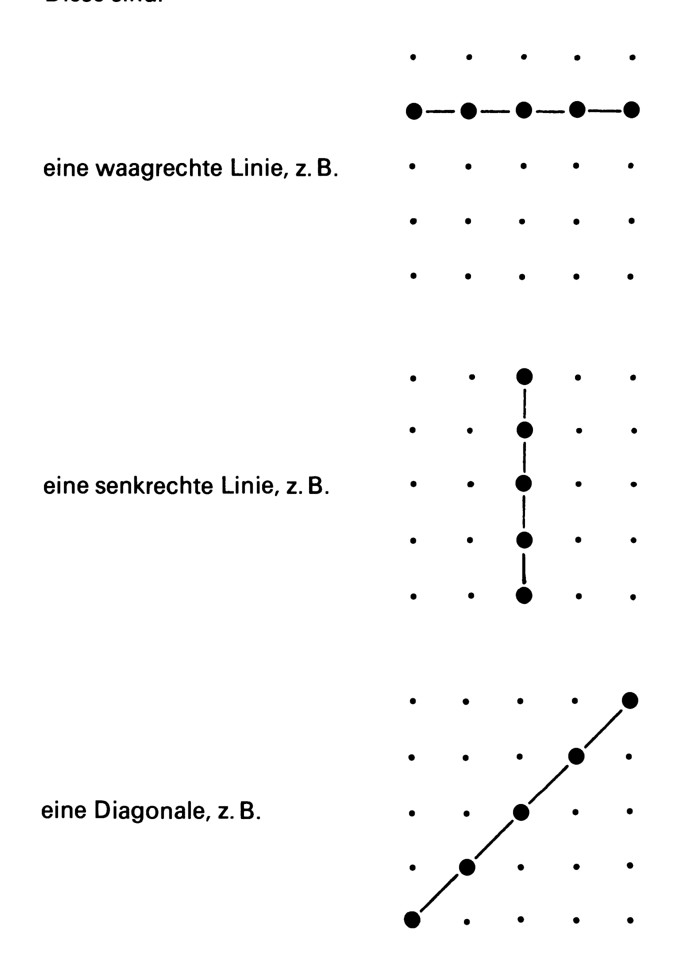
Qui Vive

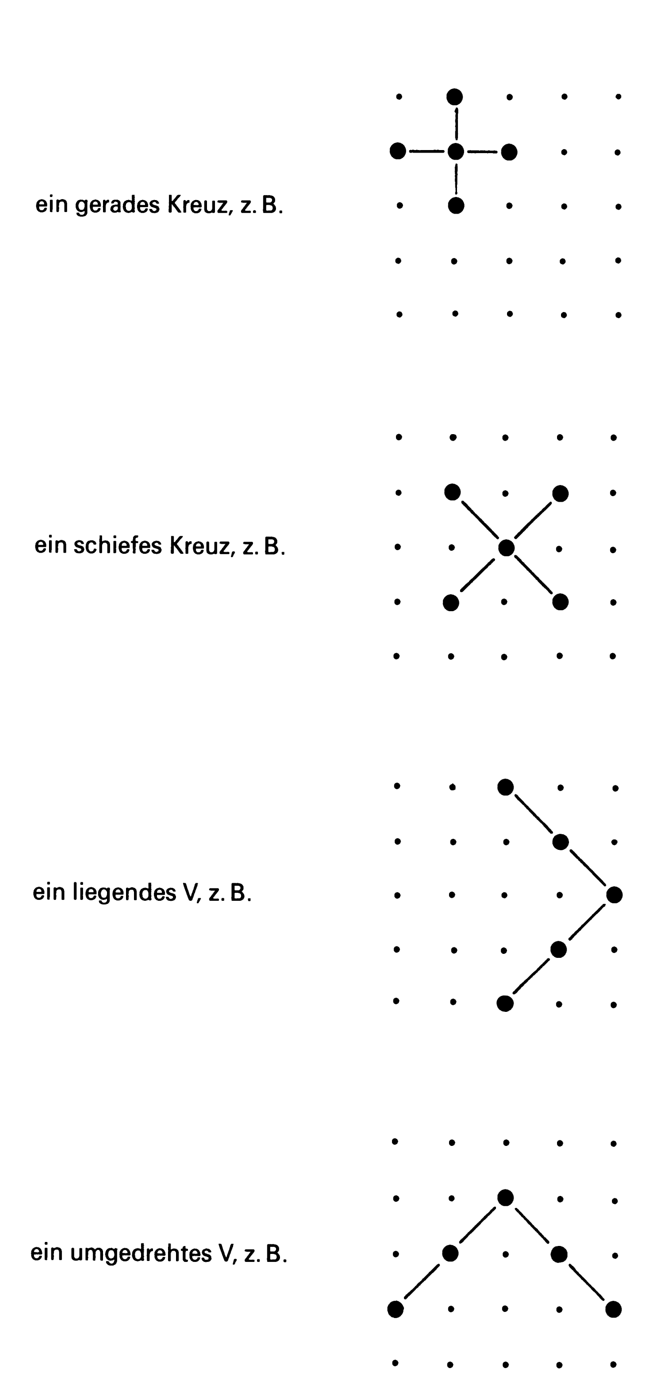
Auf Qui Vive sein, heißt wachsam zu sein und aufzupassen. Dieses Spiel heißt Qui Vive, weil man, um es erfolgreich zu spielen, dauernd auf Gewinnstellungen achten muß. Die Regeln sind einfach, sehr im Gegensatz zum Programm, das sie enthält.

In der Tat stellt es eine ziemliche Herausforderung für den Programmierer dar.

Qui Vive wurde von Eugene de Wolf erfunden und wird auf einem quadratischen 5 mal 5 Brett gespielt. Jeder Spieler hat fünf Steine und muß versuchen, sie in eine von sieben symmetrischen Mustern anzuordnen.

Diese sind:




ein gerades Kreuz, z. B.
ein schiefes Kreuz, z. B.
ein liegendes V, z. B.
ein umgedrehtes V, z. B.

Am Anfang des Spieles ist das Brett leer, Sie und der Computer sind abwechselnd an der Reihe, um Steine zu plazieren. Die Felder auf dem Brett sind so beschriftet:

⑤	1	2	3	4	5
④	6	7	8	9	10
③	11	12	13	14	15
②	16	17	18	19	20
①	21	22	23	24	25
	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ

Um einen Stein auf irgendein bestimmtes Feld zu legen, brauchen Sie nur die entsprechenden Koordinaten einzugeben:

Zum Beispiel, für Feld 21 geben Sie A1 ein, für Feld 12 B3, etc. Ihre Steine werden durch einen Ring angezeigt, die des Computers durch einen Kreis. Wenn alle zehn Steine auf dem Spielfeld plaziert sind, und keiner gewonnen hat, geht das Spiel weiter, indem Sie und der Computer abwechselnd ziehen, um eine Gewinnstellung zu erreichen. Um einen Ihrer Steine zu ziehen, geben Sie erst die Koordinaten seiner jetzigen Position ein, und dann die der Position, wo Sie ihn hinstellen wollen.

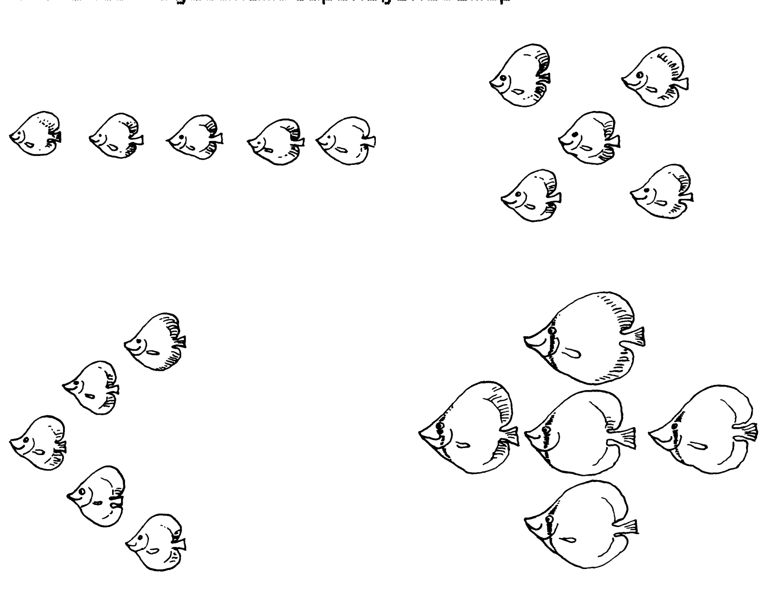
Sie müssen auf Möglichkeiten achten, selbst eine Gewinnstellung zu schaffen, aber gleichzeitig sicherstellen, daß Sie jeglichen Gewinnzug des Computers verhindern.

Dieses Programm ist eine der interessantesten in diesem Buch. Es enthält eine Liste, nicht nur von den 42 möglichen Gewinnstellungen, sondern auch von den über 1ØØ besonders starken Stellungen, in denen ein Spieler die Möglichkeit hat, eine von zwei Gewinnstellungen zu schaffen. Während des Spiels kontrolliert der Computer die Spielsituation und entscheidet, welches der bestmögliche Zug ist.

Wir empfehlen Ihnen, dieses Spiel auf einem normalen Schachbrett gegen einen Ihrer Freunde zu spielen, bevor Sie mit dem Computer anfangen. Sie werden schnell sehen, was für ein herrliches Spiel es ist.

```
10 rem * c-64 qui vive *
20 poke646,13
30 dim sf(41,4),so(41)
40 dim d(107,1),df(15,8)
60 dim cc(4,1),b(24),c(24)
70 goto 3000
100 rem * punkte von df berechnen *
110 gv=0:p1=0:p2=0:c1=0:c2=0
120 d1=b(df(nf,0)):d2=b(df(nf,1))
130 if d1=1 or d2=1 then p1=1
140 if d1=10 or d2=10 then c1=10
150 d1=b(df(nf,2)):d2=b(df(nf,3))
160 if d1=1 or d2=1 then p2=1
170 if d1=10 or d2=10 then c2=10
180 gv=b(df(nf,4))+b(df(nf,5))+b(df(nf,6
))
190 p=gv+p1+p2+c1+c2
200 if gv=30 and (c1=0 or c2=0) then p=p
-10
210 if gv=3 and(p1=0 or p2=0) then p=p-1
220 return
300 rem * einzelfiguren (sf) initialisie
ren *
310 for i=0 to 41:read x$
320 for j=1 to 5
330 sf(i,j-1)=asc(mid$(x$,j,1))-65
340 next j,i:return
350 rem v (av><)
360 data uqmsy,plhnt,kgcio
370 data agmie,flrnj,kqwso
380 data agmqu,bhnrv,ciosw
390 data eimsy,dhlrx,cgkqw
400 rem schiefes kreuz
410 data acgkm,bdhln,ceimo
420 data flhpr,gimqs,hjnrt
430 data kmquw,lnrvx,moswy
440 rem plus (+)
450 data bfghl,cghim,dhijn
460 data gklmq,hlmnr,imnos
470 data lpqrv,mqrsw,nrstx
480 rem diagonalen (/)
490 data agmsy,eimqu
500 rem spalten (i)
510 data afkpu,bglqv,chmrw
520 data dinsx,ejoty
530 rem reihen (-)
540 data abcde,fghij,klmno
550 data pqrst,uvwxy
555 rem
560 rem * doppelfiguren initialisieren *
570 for i=0 to 107 step 12:read x$
```

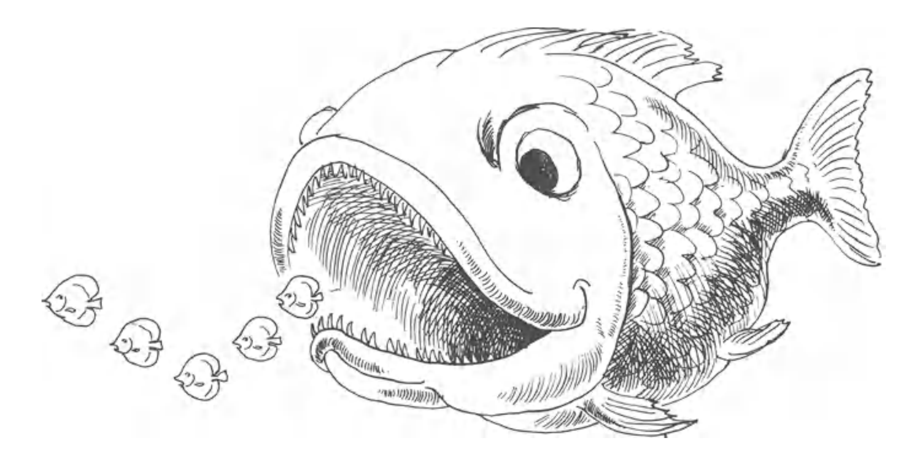
```
580 for j=0 to 11:for k=0 to 1
590 l=2*j+k+1:d(i+j,k)=asc(mid$(x$,l,1))
-49
600 next k,j,i
610 return
620 data "171:1a1c1e1m1o1p2>2`2b2j"
630 data "393<3=3?3g474:4=4?4a4g4o"
640 data "4p5`5b5d5j696<6c6e6m7=7a"
650 data "7c7i7o7p8>8b8d8j9?9e9k:?"
660 data ":a:e:k:o:p;>;`;d;j<=<c<i"
670 data "=g=i=o>f>h>j?g?k?p`f`j`l"
680 data "agaiakamaoapbhbjbncicmcp"
```



```
690 data "djdldnekemeofrfwgsgwhthw"
700 data "irixjsjxktkxlrlymsmyntny"
790 rem
800 rem * doppelfiguren berechnen *
810 tc=4:tn=0
820 for i=0 to 4:c(sf(d(df,0),i))=1:next
830 for i=0 to 4:sf=sf(d(df,1),i)
840 if c(sf)=1 then 860
850 df(nf,tn)=sf:tn=tn+1:goto 870
860 df(nf,tc)=sf:tc=tc+1:c(sf)=0
870 next i
880 for i=0 to 4:sf=sf(d(df,0),i)
890 if c(sf)=1 then df(nf,tn)=sf:tn=tn+1
:c(sf)=0
```

```
900 next i:return
1000 rem * ausgabe des spielfeldes *
1005 print spc(12)"▣   abcde▣"
1010 for i=0 to 4:print spc(12)5-i;:for
j=0 to 4
1020 f=i*5+j
1030 if b(f)=0 then print ".";
1040 if b(f)=1 then print "W";
1050 if b(f)=10 then print "Q";
1060 next j:print 5-i:next i
1070 print spc(12)"▣   abcde▣"
1080 return
1100 rem * besitzer sf bestimmen *
1110 om=-1:wm=-1
1120 for sf=0 to 41
1130 p=b(sf(sf,0))+b(sf(sf,1))+b(sf(sf,2
))+b(sf(sf,3))+b(sf(sf,4))
1140 if p=0 or p=10 or p=20 or p=30 then
 so(sf)=10:goto 1240
1150 if p<>40 then 1190
1160 gosub 1250
1170 gosub 1290
1180 wm=1:sf=41:goto 1240
1190 if p>20 or p<3 or p=11 or p=12 then
 so(sf)=0:goto 1240
1200 so(sf)=1
1210 if p=4 and om=-1 then gosub 1250
1220 if p=14 then gosub 1290
1230 if p=5 then sf=41
1240 next sf:return
1250 rem * obligatorischen zug (om) fest
legen *
1260 for i=0 to 4
1270 if b(sf(sf,i))=0 then om=sf(sf,i)
1280 next i:return
1290 rem * festgenagelten stein festlege
n *
1300 for i=0 to 4
1310 if b(sf(sf,i))<>10 then 1350
1320 for j=0 to cc-1
1330 if sf(sf,i)=cc(j,0) then cc(j,1)=1
1340 next j
1350 next i
1360 return
1400 rem * besitzer df bestimmen *
1410 nc=0:np=0:nf=0
1420 for i=0 to 15
1430 df(i,8)=-1
1440 df(i,7)=0
1450 next i
1460 for df=0 to 107
```

```
1470 o1=so(d(df,0)):o2=so(d(df,1))
1480 if o1<>o2 or o1=0 or o2=0 then 1600
1490 gosub 800
1500 gosub 100
1510 if p=4 or p=14 then df(nf,7)=1:goto
 1540
1520 if o1=1 or p<=df(nf,8) then 1600
1530 df(nf,7)=10
1540 df(nf,8)=p
1550 nf=0
1560 for i=0 to 15
1570 if df(i,8)>=df(nf,8) then 1590
1580 if df(i,7)=0 or df(i,7)=10 then nf=
1
```



```
1590 next i
1600 next df
1610 return
1800 rem * strategischen wert eines zugs
 bestimmen *
1810 if wm=1 then 1960
1820 mc=0:n2=0
1830 for nf=0 to 15
1840 if df(nf,7)=0 then 1900
1850 gosub 100
1860 if df(nf,7)=1 then 1890
1870 if p=mc then n2=n2+1
1880 if p>mc then mc=p:n2=1
1890 if p=4 then nf=15
1900 next nf
1910 if pp=-1 then 1960
1920 if p=4 then return
```

```
1930 if mc<mp then return
1940 if mc=mp and n2<n1 then return
1950 if mc=mp and n2=n1 and rnd(0)<.5 th
en return
1960 mp=mc:n1=n2
1970 pp=cp:pt=ct
1980 return
2000 rem * zug des spielers *
2010 if pc<5 then 2050
2020 print "welchen stein moechten sie z
iehen";
2030 gosub 2130:pt=x
2040 if b(pt)<>1 then print"nicht moegli
ch":goto 2020
2050 print"wohin moechten sie ziehen";
2060 gosub 2130:pp=x
2070 if b(pp)<>0 then print "nicht moegl
ich":goto 2050
2080 if pc=5 then b(pt)=0
2090 if pc<5 then pc=pc+1
2100 b(pp)=1
2110 gosub 1000
2120 return
2130 rem * eingabe *
2140 input x$:if len(x$)<>2 then 2190
2150 l$=left$(x$,1):d$=right$(x$,1)
2160 if l$<"a" or l$>"e" or d$<"1" or d$
>"5" then 2190
2170 x=asc(l$)-5*val(d$)-40
2180 return
2190 print "falsche eingabe,":print"vers
uchen sie es nochmal";:goto2140
2200 rem * zug des computers *
2210 if cc<5 then pt=cc:cc=cc+1:goto 227
0
2220 print spc(10)"ich nehme "chr$(65+pt
-int(pt/5)*5);5-int(pt/5)
2230 b(pt)=0
2240 for ci=0 to 4
2250 if cc(ci,0)=pt then pt=ci:ci=4
2260 next ci
2270 print spc(10)"ich setze ihn auf "ch
r$(65+pp-int(pp/5)*5);5-int(pp/5)
2280 b(pp)=10:cc(pt,0)=pp
2290 gosub 1000
2300 return
3000 rem * hauptprogramm *
3010 print spc(12)"bitte warten"
3020 gosub 300
3030 gosub 560
3060 cc=0:pc=0
```

```
3070 pp=12:goto 3300
3080 for i=0 to 4:cc(i,1)=0:next
3090 gosub 1100
3100 if p=5 then end
3110 if wm=1 then 3130
3120 gosub 1400
3130 mp=0:n1=0:pp=-1
3140 bf=0:ef=24
3150 if om<>-1 then bf=om:ef=om
3160 for cp=bf to ef
3170 if b(cp)<>0 then 3290
3200 b(cp)=10
3210 if cc<5 then gosub 1800:goto 3280
3220 for ci=0 to 4
3230 if cc(ci,1) then 3270
3240 ct=cc(ci,0):b(ct)=0
3250 gosub 1800
3260 b(ct)=10
3270 next ci
3280 b(cp)=0
3290 next cp
3300 gosub 2200
3310 if wm=1 then end
3320 gosub 2000
3330 print spc(12) "bitte warten"
3340 goto 3080
```

K. Z. G.

Das Wesen des Gedächtnisses ist sicherlich eines der interessantesten Probleme der Psychologie. Sich an etwas erinnern wurde einmal definiert als "in jetzigen Reaktionen Zeichen früher erlernter Reaktionen zeigen". Aber warum erinnern wir uns an manche Dinge und vergessen andere?

Beispielsweise erkennen wir jemanden wieder, den wir seit Jahren nicht mehr gesehen haben, oder uns fällt eine Melodie ein, von der wir nur ein paar Noten gehört haben. Andrerseits finden wir es äußerst schwierig, zum Beispiel eine Telefonnummer zu behalten. Es gibt eigentlich zwei verschiedene Gedächtnistypen: Das Kurzzeitgedächtnis (KZG), und das Langzeitgedächtnis (LZG).

Wenn Sie mehr über dieses faszinierende Thema wissen wollen, empfehlen wir Ihnen das Buch "Introduction to Psychology" von Hilgard und Atkinson.

Dieses Spiel testet Ihr Kurzzeitgedächtnis. Spielen Sie es mit Freunden und Familie, schauen Sie, wer sich am meisten merken kann. Sie werden folgendes "Menü" auf dem Schirm sehen:

MOECHTEN SIE SPIELEN MIT

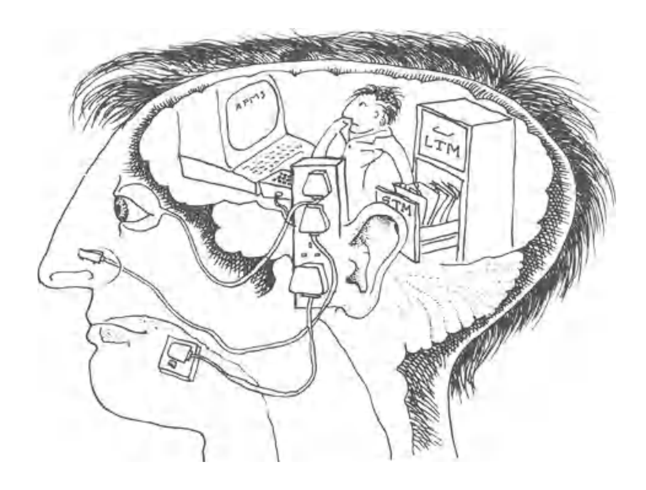
1. BUCHSTABEN?
2. ZAHLEN?
3. NUR Ø ODER 1?
4. ODER MOECHTEN SIE AUFHOEREN?

Jetzt wählen Sie, indem Sie 1, 2, 3 oder 4 eingeben. Nehmen wir einmal an, Sie hätten 1 eingegeben, um mit Buchstaben zu spielen. Ein Buchstabe wird jetzt auf dem Schirm erscheinen, aber nur für ganz kurze Zeit. Sie sollen diesen Buchstaben eingeben. Der Computer wird Ihnen dann zwei Buchstaben zeigen, die Sie eingeben sollen, dann drei, und so weiter. Offensichtlich wird es mit zunehmender Anzahl der Buchstaben immer schwieriger, sich diese alle zu merken.

Welches ist der längste Ausdruck, den Sie behalten können? Können Sie unseren Rekord von acht brechen?

```
10 rem * c-64 k.z.g. *
20 print "▒▒moechten sie spielen mit"
30 print "▒1:buchstaben?"
40 print "▒2:zahlen?"
50 print "▒3:nur 0 und 1?"
60 print "▒4:oder moechten sie aufhoeren
?"
70 get a$
80 on val(a$) goto 100,110,120,620
90 goto 70
100 a=26:b=65:goto 200
```

```
110 a=10:b=48:goto 200
120 a= 2:b=48
200 rem * string bilden *
205 input"▣fertig???";k$
209 if k$=" " then 209
210 c=1:tt=300
220 t=ti:ta=t
230 a$=""
240 for d=1 to c
250 b$=chr$(b+rnd(0)*a)
260 print b$;
270 a$=a$+b$
280 for e=1 to tt:next
290 print "▮▮ ";
300 next
310 print
400 rem * antwort *
410 input "antwort";an$
420 if an$=a$ then 600
430 print "▣falsch!"
440 print "zahl der versuche";c
450 print "es war:";a$
460 print "zeit:";str$(int((ta-t)/.6)/10
0);" sekunden"
500 print "nochmal (j/n)?"
510 get a$
520 if a$="j" then 200
530 if a$<>"n" then 510
540 goto 10
600 c=c+1:ta=ti
610 goto 230
620 end
```



Eins zu Fünf

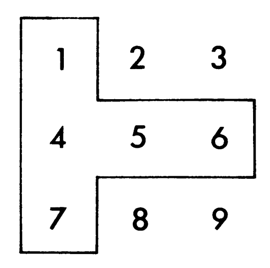
Dieses kleine Spielchen, das wir das erstemal auf einem Casio Rechner sahen, verlangt zur Lösung einiges an logischem Überlegen. Es wird auf einem Brett mit neun Feldern gespielt, das so aufgeteilt ist:

```
1   2   3

4   5   6

7   8   9
```

Ein Linienkreuz durch irgendein Feld wird insgesamt fünf Felder kreuzen. Die zum Beispiel durch Feld 4 gezogenen Linien würden auch die Felder 1, 5, 6 und 7 berühren.



Die Felder auf dem Brett sind mit Zufallszahlen zwischen Ø und 5 belegt. Wenn Sie eines dieser Felder eingeben (und die auf dem Schirm gezeigte Numerierung benutzen), wird die Zahl auf dem betreffenden Feld und alle Zahlen auf der horizontalen und vertikalen Linie durch das betreffende Feld um 1 erhöht (außer 5; 5 wird zu Ø). Zum Beispiel, wenn wir hätten:

```
2   2   4

1   5   2

1   5   4
```

und Sie gäben 4 ein, würde das Brett wechseln zu:

3	2	4
2	0	3
2	5	4

Das Puzzle ist gelöst, wenn das Brett so aussieht:

0	0	0
0	0	0
0	0	0

```
10 rem * c-64 eins zu fuenf *
15 d$="▮▮▮▮▮▮▮▮▮▮▮▮"
20 m=0
30 dim a%(2,2)
40 goto 1000: rem hauptprogramm
100 rem * bildschirm *
110 print "▣"
120 print d$ "  ┌───┬───┬───┐"
130 print d$ "  ] ] ] ]  q w e"
140 print d$ "  ├──[──[───┤"
150 print d$ "  ] ] ] ]  a s d"
160 print d$ "  ├──[──[───┤"
170 print d$ "  ] ] ] ]  z x c"
180 print d$ "  └───┴───┴───┘"
190 return
200 rem * brett zeichnen *
210 for a=1 to 11
220 i=9*rnd(0)+1
230 gosub 300:rem ziehen
240 gosub 500:rem brett drucken
250 next
260 return
300 rem * zug ausfuehren *
310 y=int((i-1)/3)
```

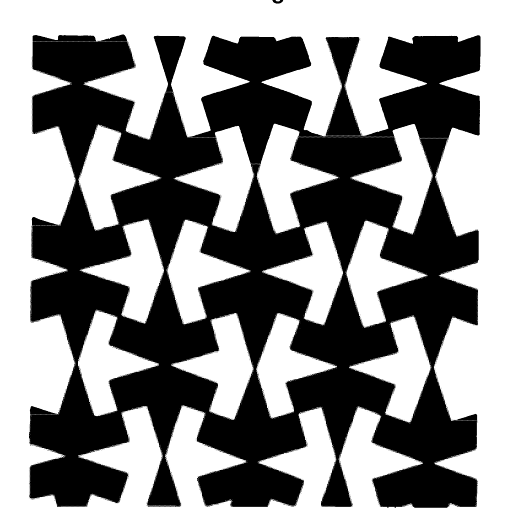
```
320 x=i-1-3*y
330 for z=0 to 2
340 a%(x,z)=a%(x,z)+1
350 a%(z,y)=a%(z,y)+1
360 next
370 a%(x,y)=a%(x,y)-1
380 wf=1
390 for p=0 to 2
400 for q=0 to 2
410 if a%(p,q)=6 then a%(p,q)=0
420 wf=wf and (a%(p,q)=0)
430 next: next
440 return
500 rem * brett ausdrucken *
510 print "[illegible]" d$
520 for q=0 to 2
530 print d$ "[illegible]";
540 for p=0 to 2
550 print "[illegible]";right$(str$(a%(p,q)),1);
560 next p
570 print
580 next q
590 return
600 rem * eingabe *
610 get a$:if a$="" then 610
620 for a=1 to 9
630 if a$=mid$("qweasdzxc",a,1) then i=a
:a=9
640 next
650 m=m+1
660 return
1000 rem * hauptprogramm *
1010 gosub 100: rem bildschirm
1020 gosub 200: rem brett ausgeben
1030 gosub 600: rem eingabe
1040 gosub 300: rem zug ausfuehren
1050 gosub 500: rem brett ausdrucken
1060 if wf=0 then 1030
1070 print "[illegible]"
1080 print "sie haben es geschafft in";m
;" zuegen"
```

Escher

Weil das Werk des holländischen Grafikkünstlers M. C. Escher (19Ø2–1972) sich auf Symmetrie und mathematische Formen stützt, spricht es besonders den Computerprogrammierer an.

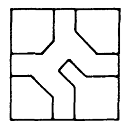
Jetzt können Sie mit Hilfe des Computers Ihre eigenen Muster erzeugen, aufgebaut mit denselben Regeln, wie jene, die von Escher selbst verwendet wurden.

Betrachten Sie das folgende Muster:

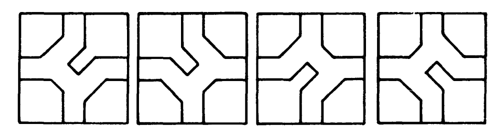


Es basiert auf einem Mosaik im Alhambra Palast in Spanien. Wenn Sie genau hinschauen, können Sie sehen, daß alle Elemente die gleiche Form haben, und daß sie so angeordnet sind, daß das "Negativ" der weißen Elemente das gleiche Muster um 9Ø Grad gedreht zeigt.

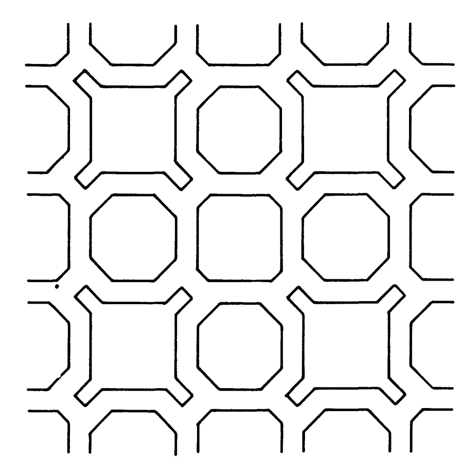
Escher studierte solche Muster und erfand ein Spiel, das darauf aufbaut. Er entwarf ein gemustertes Element wie dieses



das er durch 4 Positionen rotierte



und so anordnete, daß sie ein fortlaufendes Muster formten.



Dieses Programm tut etwas sehr ähnliches. Nachdem Sie ein Grundelement eingegeben haben, rotiert und verschiebt der Computer es, um den Schirm mit Ihrem eigenen "Escher" zu füllen.

Am Anfang des Spieles fragt der Computer Sie nach der Größe Ihres Grundelementes. Wenn Sie beispielsweise 4 eingeben, wird Ihr Grundelement 4 mal 4 Kästchen enthalten. Nun füllen Sie Ihr Quadrat, Reihe für Reihe, mit den in den Zeilen 3000 bis 3200 des Programmausdrucks gezeigten Grafiksymbolen. Wenn das Quadrat gefüllt ist, fragt der Computer nach der Richtung, in welcher Sie die Quadrate abgebildet haben möchten. Sie geben N, S, O oder W ein für Norden, Süden, Osten und Westen. Der Computer wird Sie nach allen benötigten Richtungen fragen.

Mit diesem Programm sind Sie imstande, eine fast unendliche Anzahl eindrucksvoller Muster zu erzeugen. Geben Sie das Programm ein, und sehen Sie, was dabei herauskommt.

```
10 rem * c-64 escher spiel *
20 mk=20
30 print "▒"
40 dim sy(mk,3)
50 for cc=0 to mk
60 for r1=0 to 3
70 read sy(cc,r1)
80 next
90 next
100 input"groesse des grundelementes";s
110 sx=int(39/s)+1
115 sy=int(23/s)+1
120 dim ds(sx,sy),bs(s,s),ri(s,s),sc(s*s
x,s*sy)
130 print "▒";
```

```
140 for r=0 to s-1
150 print "reihe";r+1;tab(7);
160 a$=""
170 input a$
180 a$=left$(a$+"             ",s)
190 for k=0 to s-1
200 bs(r,k)=asc(mid$(a$,k+1,1))
210 next k
220 next r
230 print
240 print "richtung der elemente (n,s,w,
o)"
250 print "jede reihe"sx" richtungen"
260 print
270 for b=0 to sy-1
280 print "reihe";b+1;tab(7);
290 input a$
300 a$=left$(a$+"                            "
,sx)
310 for a=0 to sx-1
320 in$=mid$(a$,a+1,1)
330 ds(a,b)=-((in$="n")+2*(in$="o")+3*(i
n$="s")+4*(in$="w"))-1
340 if ds(a,b)<0 then ds(a,b)=0
350 next a
360 next b
370 print
380 print "{bild berechnen}"
390 for r=0 to s-1
400 for k=0 to s-1
410 for r1=3 to 0 step -1
420 for cc=mk to 0 step -1
430 if bs(k,r)=sy(cc,r1) then c2=cc:r2=r
1:cc=0:r1=0
440 next
450 next
460 bs(k,r)=c2
470 ri(k,r)=r2
480 next
490 next
1000 print "{graphik gefunden}"
1010 for r=0 to s-1
1020 for k=0 to s-1
1030 for b=0 to sy-1
1040 for a=0 to sx-1
1050 on ds(a,b)+1 goto 1060,1070,1080,10
90
1060 x=k:y=r:goto 1100
1070 x=s-r-1:y=k:goto 1100
1080 x=s-k-1:y=s-r-1:goto 1100
1090 x=r:y=s-k-1
```

```
1100 r1=(ds(a,b)+ri(r,k))and 3
1110 sc(x+a*s,y+b*s)=sy(bs(r,k),r1)
1120 next
1130 next
1140 next
1150 next
2000 rem * escher zeichnen *
2010 print "▒";
2020 for y=0 to 23
2030 for x=0 to 39
2040 if sc(x,y)>255 then print "▒";:goto
 2060
2050 print "▒";
2060 print chr$(sc(x,y)and 255);
2070 next
2080 next
2090 print "}";
2100 goto 2100
3000 data  32, 32, 32, 32:rem"  "
3010 data 165,163,167,164:rem"| "
3020 data 212,197,217,210:rem"T"
3030 data 199,196,200,198:rem"G"
3040 data 194,195,221,192:rem"B"
3050 data 180,183,170,175:rem"| "
3060 data 181,184,182,185:rem"▌ "
3070 data 161,418,417,162:rem"█ "
3080 data 205,206,205,206:rem"M"
3090 data 203,202,213,201:rem"K"
3100 data 189,173,176,174:rem"┘ "
3110 data 207,208,186,204:rem"O"
3120 data 177,171,178,179:rem"┴"
3130 data 169,223,425,479:rem"  "
3140 data 190,188,172,187:rem"▀ "
3150 data 209,209,209,209:rem"Q"
3160 data 214,214,214,214:rem"V"
3170 data 215,215,215,215:rem"W"
3180 data 219,219,219,219:rem"["
3190 data 166,422,166,422:rem"▒"
3200 data 191,447,191,447:rem"▀▄"
```

Genie am Werk

Spielen Sie dieses kleine Spielchen, um herauszufinden, wie gut Sie mathematisch denken können. Oder lassen Sie gleich die ganze Familie mitspielen und finden Sie heraus, wer das Genie ist!

Der Computer gibt sieben Zahlen und eine größere "Zielzahl" an. Sie wählen zwei dieser Zahlen und eine von den vier Grundrechenarten:

Addition	+
Subtraktion	−
Multiplikation	*
Division	/

Der Computer wird dann diese Rechnung ausführen. Beispielsweise, wenn Sie 7 + 18 eingegeben haben, berechnet der Computer 7 + 18 = 25 und fragt dann:

LASSEN SIE ES SO?

Antworten Sie mit N, dann werden die Zahlen 7 und 18 in der ursprünglichen Liste ersetzt durch 25.

Sie können dann eine weitere Rechnung angeben. Ihr Ziel ist, mit einer Zahl aufzuhören, die sehr nahe an die Zielzahl herankommt.

Geben Sie J ein, wenn Sie so nahe wie nur möglich herangekommen sind. Eine zusätzliche Erschwernis ist es, daß Sie all dies innerhalb einer bestimmten Zeit tun sollen.

```
10 rem * c-64 genie am werk *
20 t=0:x=0:r=0
30 dim d(7)
40 goto 160
50 input "rechnung";f$
60 for n=1 to len(f$)
70 e=asc(mid$(f$,n))
80 if e=42 or e=43 or e=45 or e=47 then
110
90 next
100 goto 50
110 a=val(mid$(f$,1,n-1))
120 b=val(mid$(f$,n+1))
130 t$=mid$(f$,n,.1)
140 if val(f$)<>int(val(f$)) then 50
150 goto 310
160 print "♥";
170 for i=1 to 7
180 c=int(10*rnd(0)+1)
190 if int(4*rnd(0))=3 then c=int(4*rnd(
0)+1)*25
```

```
200 d(i)=c
210 next
220 f=int((9*rnd(0)+1)*100)
230 print "[illegible]versuchen sie diese zahl zu
 erreichen:    ";f
240 print
250 print "sie haben diese zahl erhalten
"
260 for i=1 to 7
270 print str$(d(i));
280 next
290 print:print
300 goto 50
310 for i=1 to 7
320 if d(i)=a then 350
330 next i
340 goto 460
350 d(i)=0
```



```
360 for i=1 to 7
370 if d(i)=b then 400
380 next i
390 goto 460
400 gosub 600
405 print f$;"=";d(i)
410 print "lassen sie es so";
420 z$="":input z$
430 if z$="" then 230
440 if left$(z$,1)="j" then 490
450 goto 230
460 print "diese zahlen haben sie nicht
erhalten,  sie bekommen keine punkte"
470 x=0
480 goto 510
490 print "sie haben:";d(i);" auf";f
500 x=5-abs(f-d(i))
510 if x<0 then x=0
520 if x=5 then print "gut gemacht"
530 t=t+x
540 r=r+1
550 print "sie bekommen jetzt";x;" punkt
e"
560 print "sie haben jetzt";t;" punkte,
in";r;" runden"
570 input "nochmal";z$
580 if left$(z$,1)="j" then 160
590 end
600 if t$="+" then d(i)=a+b:return
610 if t$="-" then d(i)=a-b:return
620 if t$="*" then d(i)=a*b
630 if t$="/" then d(i)=a/b
640 return
```

Haifischjagd

Seit fünf Tagen treiben Sie nun schon in Ihrem Boot umher und durchsuchen die Weltmeere. Plötzlich sehen Sie ein leichtes Plätschern auf der spiegelglatten Meeresoberfläche – da ist er, Ihr Todfeind, der Schrecken des Meeres – der Haifisch.

Zu Beginn des Spieles erfahren Sie, wie Sie Ihr Boot bewegen können. Ihr Echolot wird Ihnen jeweils sagen, wie dicht Sie am Hai sind – je näher Sie rankommen, desto höher wird der Ton sein, den Sie hören. Wenn Sie sich zu weit vom Tier entfernt haben, wird angezeigt:

SCHADE, SIE GINGEN ZU WEIT WEG

Wenn Sie glauben, nahe genug zu sein, können Sie einen Schuß auf den Haifisch abgeben: Drücken Sie auf die Leertaste. Falls Sie getroffen haben, erscheint die Meldung:

DAS WAR'S! GUT GEMACHT

Unglücklicherweise haben Sie nur 2ØØ Energieeinheiten zu Ihrer Verfügung. Sowohl das Bewegen des Bootes als auch die Abgabe eines Schusses kosten Energie. Wenn Sie alle Energie verbraucht haben, zeigt der Computer an:

SIE HABEN KEINE ENERGIE MEHR

Zum Schluß erhalten Sie Informationen über Ihre Fähigkeiten als Haifischjäger.

```
10 rem * c-64 haijagd *
20 dim x$(2),y$(2),tu(3,8),co$(2)
30 d$="[illegible]"
35 sid=54272
40 for i=0 to 23:pokesid+i,0:next:pokesi
d+24,15:pokesid+5,15:pokesid+6,255
50 b$="[illegible]"
60 ng=0:tr=0:mr=0
70 for re=0 to 2
80 read x$(re),y$(re)
90 next
100 data 1,f,"","",r,b
110 rem tune
120 for tu=1 to 3
130 for no=1 to 8
140 read tu(tu,no)
150 next no,tu
160 data 40,60,40,60,40,50,60,60
170 data 30,35,40,45,50,55,60,60
180 data 35,32,30,27,25,22,20,20
```

```
190 co$(0)="schade, sie gingen zu weit w
eg"
200 co$(1)="das war's!          gut gemac
ht"
210 co$(2)="sie haben keine energie mehr
"
230 goto 3000
300 rem * ton *
310 poke sid+4,17:pokesid+1,tn
320 for t=1 to tm
330 for dl=1 to 10:next dl
340 next t
360 return
```



```
1000 rem * initialisieren *
1010 ng=ng+1
1020 sx=int(70*rnd(0))-35
1030 sy=int(70*rnd(0))-35
1040 di=abs(sx)+abs(sy)
1050 md=di:sd=di
1060 en=200:sh=0:hi=0:ns=0
1070 vx=0:vy=0
1080 return
1100 rem * eingabe der geschwindigkeit *
1110 get k$
1120 vx=4*((k$="▐▌")-(k$="▌▌"))
1130 vy=4*((k$="▣")-(k$="◙"))
1140 sh=(k$=" ")
1150 return
1300 rem * entfernung etc. berechnen *
1310 sx=sx-vx
1320 sy=sy-vy
1330 en=en+(vx<>0)+(vy<>0)-1
1340 if en<=0 then en=0:goto 1460
1350 rem hai bewegt sich weg
1360 sx=sx+sgn(sx)
1370 sy=sy+sgn(sy)
1380 di=abs(sx)+abs(sy)
```

```
1390 if di<md then md=di
1400 if di>100 then 1460
1410 if not sh then 1460
1420 ns=ns+1
1430 en=int(en-.5*di)
1440 if en<=0 then en=0:goto 1460
1450 hi=int(di*rnd(0))<=1
1460 return
1500 rem * ausgabe *
1510 gosub 2000:rem blank
1520 print left$(d$,8);tab(sgn(vx)+19);x
$(sgn(vx)+1);
1530 print left$(d$,8+sgn(vy));tab(19);y
$(sgn(vy)+1);
1540 if sh then tm=2:tn=10:gosub 300:got
o 1560
```



```
1550 tm=1:tn=140-di:gosub 300
1560 print "▤◙";tab(14);en;"▐▌ ";
1570 return
1600 rem * bildschirm *
1620 print "▤◙";spc(7)"energie:"en"ergs"
1640 print left$(d$,15);"bewegen:  curso
r steuertasten verwenden"
1650 print "◙schiessen: space-taste drue
cken"
1660 gosub 2000:rem blank
1670 print left$(d$,8);"richtung:"
1680 print "▤▥◙";"<return> druecken um a
nzufangen";
1685 input an$
1710 return
1800 rem * ende des spiels *
1810 print "▤◙";spc(5);co$(ca-1)
```

```
1820 for no=1 to 8
1830 tn=tu(ca,no):tm=10:gosub 300
1840 next
1845 poke sid+4,0
1850 print "■■sie fingen an bei:";sd;tab
(20);"m"
1860 print "zahl der schuesse:";ns
1870 print "sie naeherten sich bis auf:"
;md;tab(20);"m"
1880 if hi then print "sie trafen in:";d
i;tab(20);"m"
1890 if hi then re=50+en/4+sd/10:goto 19
00
1895 re=50-di/2
1900 re=int(re)
1910 if re>100 then re=100
1915 if re<0 then re=0
1920 tr=tr+re
1930 if re>mr then mr=re
1940 print "■ergebnis (0-100)"
1945 print "─C──────────────────"
1950 print "zahl der spiele:";ng
1955 print "dieses spiel:    ";re
1960 print "durchschnitt:    ";int(tr/ng)
1970 print "maximum:         ";mr
1980 return
2000 rem * leerstellen *
2010 print left$(d$,7);tab(18);"    "
2020 print tab(18);"    "
2030 print tab(18);"    "
2040 return
3000 rem * hauptprogramm
3010 gosub 1000:rem initialisieren
3020 gosub 1600:rem bildschirm
3030 gosub 1100:rem eingabe
3040 gosub 1300:rem berechnen
3050 gosub 1500:rem ausgabe
3060 ca=-3*(en=0)-2*hi-(di>100)
3070 if ca=0 then 3030
3080 gosub 1800:rem ende
3090 print
3095 for i=1 to 10:get k$:next
3100 input "■ein weiteres spiel";an$
3110 if left$(an$,1)="j" then 3010
3120 end
```

Shakespeare Shuffle

Shakespeare, einer der größten Schriftsteller der Welt, und Schach, König der Spiele, sind in diesem ungewöhnlichen Puzzle vereint.

Auf einem Schachbrett sind Buchstaben angeordnet, und zwar folgendermaßen:

S	E	I	N		O	D	E
R		N	I	C	H	T	
S	E	I	N	,		D	A
S		I	S	T		D	I
E		F	R	A	G	E	.
W	I	L	L	I	A	M	
S	H	A	K	E	S	P	E
A	R	E		1	6	0	3

Eine Schachfigur, der Springer, zieht dann über das Brett, genau wie in einer normalen Schachpartie. Beim Springen von einem Feld zu einem anderen werden die Buchstaben oder Zeichen auf diesen beiden Feldern vertauscht.

Auf diese Weise wird der Text auf dem Brett durcheinandergemischt.

Der Computer wird Sie fragen

NIVEAU?

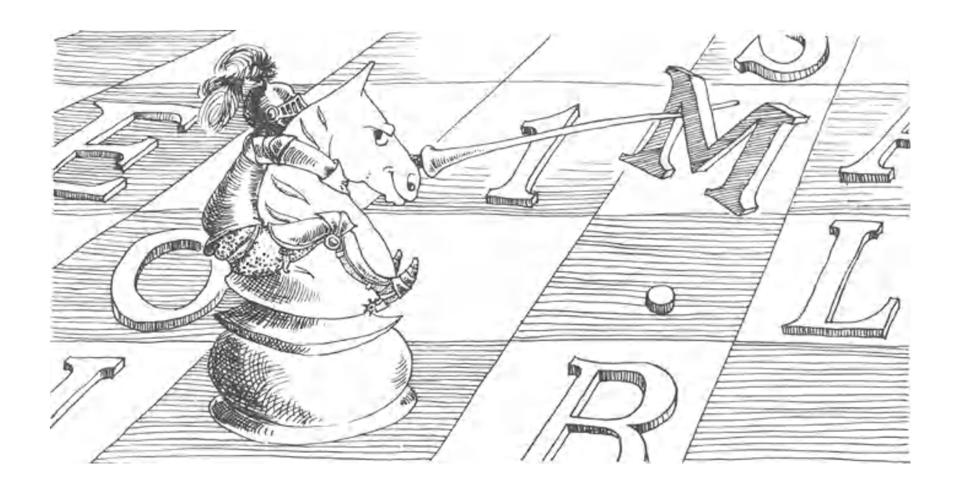
Und Sie geben die Zahl der Züge ein, die der Springer machen soll. Ganz klar, je höher diese Zahl, desto mehr wird das Brett durcheinandergeschüttelt.

Das Brett wird gezeigt, nachdem der Springer seine Züge gemacht hat. Ihre Aufgabe ist, den Text wiederherzustellen, indem Sie den Springer auf dem Brett bewegen.

Geben Sie eine Zahl zwischen 1 und 8 ein, um den Springer zu ziehen, so wie es hier gezeigt ist.

	1		2	
8				3
		♞		
7				4
	6		5	

```
10 rem * c-64 shakespeare shuffle *
15 poke646,13
20 d$="{home}{down}{down}{down}{down}{down}{down}{down}{down}{down}{down}{down}{down}"
30 dim hx(8),hy(8),b$(8,8),t$(8)
40 for k=1 to 8
50 read t$(k)
60 next
70 data "sein ode","r nicht ","sein, da"
80 data "s ist di","e frage."
90 data "william ","shakespe","are 1603"
100 for x=1 to 8
110 for y=1 to 8
120 b$(x,y)=mid$(t$(y),x,1)
130 next:next
140 for k=1 to 8
150 read hx(k),hy(k)
160 next k
170 data -1,-2,1,-2,2,-1,2,1,1,2,-1,2,-2
,1,-2,-1
```



```
180 c1$="t":c2$="█t▒"
190 xo=1:yo=1:xn=1:yn=1
200 goto 1000
300 rem * buchstabe drucken *
310 print left$(d$,y+1);tab(2*x);spc(9)
b$(x,y);
320 return
400 rem * shuffle *
410 input"▒▣level";lv
420 for du=1 to lv
430 nm=int(rnd(0)*8+1):gosub 500
440 if xo=xn and yo=yn then 430
450 xo=xn:yo=yn
460 next
470 return
500 rem * zug des springers *
510 xs=hx(nm):ys=hy(nm)
520 if xo+xs<1 or xo+xs>8 or yo+ys<1 or
yo+ys>8 then return
530 xn=xo+xs:yn=yo+ys
540 w$=b$(xn,yn)
550 b$(xn,yn)=b$(xo,yo)
560 b$(xo,yo)=w$
570 if not di then return
580 x=xo:y=yo:gosub 300
590 x=xn:y=yn:gosub 300
600 xo=xn:yo=yn
610 return
700 rem * bildschirm *
710 print"▒"
720 for x=1 to 8
730 for y=1 to 8
740 gosub 300
750 next
760 next
770 print d$;tab(16);"1 2▣"
780 print tab(14);"8     3▣"
790 print tab(17);"h▣"
800 print tab(14);"7     4▣"
810 print tab(16);"6 5"
820 return
900 rem * zug eingeben *
910 for de=1 to 200:next de
920 print left$(d$,yo+1);tab(2*xo);spc(9
) c1$
930 for de=1 to 200:next de
940 print left$(d$,yo+1);tab(2*xo);spc(9
) c2$
950 get i$
960 if i$>"" then return
970 goto 910
```

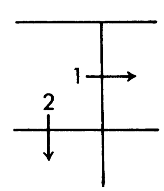
```
1000 rem * hauptprogramm *
1010 gosub 400:rem shuffle
1020 di=-1
1030 gosub 700:rem bildschirm
1040 gosub 900:rem zug eingeben
1050 nm=val(i$):if nm=0 or nm=9 then 104
0
1060 gosub 500:rem zug des springers
1070 goto 1040
```

Explosion

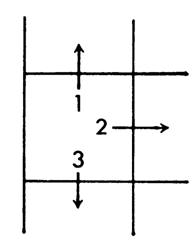
Die meisten Computerversionen von EXPLOSION erstellen ein Brett für zwei oder mehr Mitspieler. Mit diesem Programm spielen Sie gegen den Computer selbst. Sind Sie dieser Herausforderung gewachsen?

• Explosion wird auf einem Brett von 3 mal 3 oder 4 mal 4 Feldern gespielt. Der Computer wird Sie bitten, die Größe des gewünschten Brettes einzugeben.

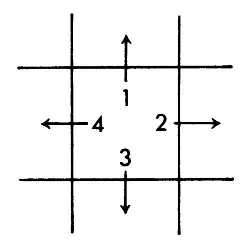
Jedes Feld auf dem Brett hat eine Kapazität, die genauso groß ist, wie die Zahl seiner Nachbarfelder. Dies bedeutet, daß Eckfelder eine Kapazität von 2 haben



Randfelder haben eine Kapazität von 3

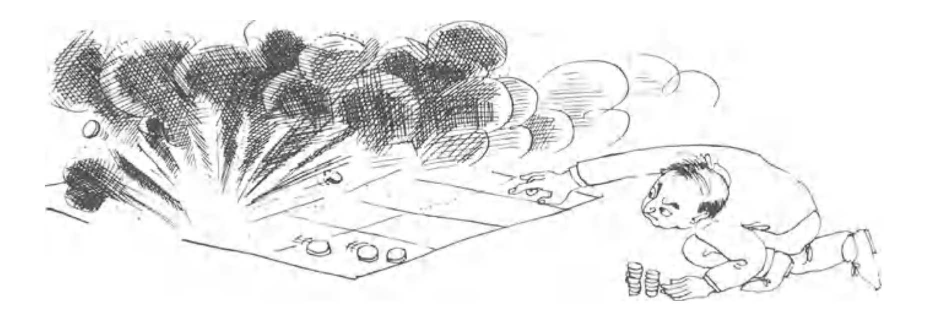


und zentrale Felder haben eine Kapazität von 4



Sie und der Computer haben Spielsteine von unterschiedlicher Farbe, die Sie, wenn Sie an der Reihe sind, auf das Brett legen. Sie dürfen dann einen Stein auf

irgendein leeres Feld setzen, oder auf irgendein Feld, auf dem bereits einer oder mehrere Ihrer Steine liegen. Ein Feld wird "explodieren", wenn die Zahl der Steine, die es enthält, die Kapazität erreicht. Ein explodierendes Feld entleert sich, seine Steine verteilen sich auf die Nachbarfelder; ein Stein pro Feld. Diese Steine werden jegliche gegnerische Steine auf diesen Feldern umwandeln. Im Spielverlauf werden die Explosionen immer größer. Am Ende wird das ganze Brett in einer Farbe explodieren. Wenn es Ihre Farbe ist, haben Sie gewonnen!



```
10 rem * c-64 explosion *
15 poke53280,0:poke53281,0
20 dim rb(5,5),sb(5,5),st(5,5)
30 d$="[illegible]"
40 goto 3000
200 rem * sb nach rb uebernehmen *
210 for x=1 to si
220 for y=1 to si
230 sb(x,y)=rb(x,y)
240 next:next
250 return
300 rem * rb nach sb uebernehmen *
310 for x=1 to si
320 for y=1 to si
330 rb(x,y)=sb(x,y)
340 next:next
350 return
400 rem * initialisieren *
410 print "[illegible]geben sie die groesse des br
ettes an"
420 input"(3 oder 4)";si
430 if si<3 or si>4 then 410
440 for x=1 to si
450 for y=1 to si
460 st(x,y)=4+(x=1)+(x=si)+(y=1)+(y=si)
470 next:next
490 print "moechten sie anfangen (j/n)";
500 input an$
510 cm=(left$(an$,1)<>"j")
600 rem * bildaufbau *
```

```
610 print "▩"
620 print " M x"
630 print "  M ";
640 for x=1 to si:print x;" ";:next:prin
t
650 for y=1 to si
660 if y=1 then print " y ";:goto 680
670 print "   ";
680 for x=1 to si:print "[———";:next:pri
nt "⊣"
690 for i=1 to 3
700 print " ";
710 if i=2 then print str$(y);:goto 730
720 print "  ";
730 for x=1 to si:print "]   ";:next:pri
nt "]"
740 next i,y
760 print "   ";
770 for x=1 to si:print "⊥———";:next:pri
nt "┘"
780 return
800 rem * zug des spielers *
810 print d$;
820 input "▩ihr zug  (x,y)      ■■■■■■■■";
mx,my
830 if mx<1 or mx>si or my<1 or my>si th
en 810
840 if rb(mx,my)<0 then 810
850 gosub 200
860 x=mx:y=my:di=-1
870 gosub 1000:rem explosion
880 if et then 2200:rem end
890 gosub 300
900 return
1000 rem * ausgabe explosion *
1010 sb(x,y)=sb(x,y)+pl
1020 ne=0
1030 if di then x1=x:y1=y:gosub 1400
1040 xp=0
1050 for y=1 to si
1060 for x=1 to si
1070 if abs(sb(x,y))<st(x,y) then 1160
1080 xp=-1
1090 ne=ne+1
1100 sb(x,y)=sb(x,y)-st(x,y)*pl
1110 if di then x1=x:y1=y:for i=1 to 999
:next i:gosub 1400
1120 x1=x:y1=y-1:gosub 1300
1130 x1=x+1:y1=y:gosub 1300
1140 x1=x:y1=y+1:gosub 1300
1150 x1=x-1:y1=y:gosub 1300
```

```
1160 next x,y
1170 et=(ne>1.5*si*si)
1180 if xp and not et then 1040
1190 return
1300 rem * verteilung auf die nachbarn *
1310 sb(x1,y1)=pl*(abs(sb(x1,y1))+1)
1320 if di and st(x1,y1)>0 then gosub 14
00
1330 return
1400 rem * feldausdruck *
1410 for i=1 to 500:next i
1420 print left$(d$,4*y1+1);tab(4*x1);
1430 print "                     ";
1440 if sb(x1,y1)=0 then 1480
1450 if pl=-1 then print "  ";
1455 if pl=1  then print"  ";
1460 print "      ";abs(sb(x1,y1));"    
        ";
1470 print " ";
1480 return
1600 rem * zug des computers *
1610 print d$;" mein zug             "
1620 be=1000
1630 for tx=1 to si
1640 for ty=1 to si
1650 if rb(tx,ty)>0 then 1720
1660 gosub 200
1670 x=tx:y=ty:di=0
1680 gosub 1000:rem ausgabe explosion
1690 if et then mx=tx:my=ty:goto 1800
1700 gosub 2000:rem bewertung
1710 if en<be or (en=be and rnd(0)<.4) t
hen be=en:mx=tx:my=ty
1720 next:next
1800 rem * tatsaechlicher zug *
1810 gosub 200
1820 x=mx:y=my:di=-1
1830 print d$;tab(8);x;",";y
1840 gosub 1000:rem ausgabe explosion
1850 if et then 2200:rem ende
1860 gosub 300
1870 return
2000 rem * bewerten *
2010 en=0
2020 for x=1 to si
2030 for y=1 to si
2040 en=en+sb(x,y)
2050 if -sb(x,y)<st(x,y)-1 then 2110
2060 en=en-2
2070 if sb(x+1,y)=st(x+1,y)-1 then en=en
+10
```

```
2080 if sb(x,y+1)=st(x,y+1)-1 then en=en
+10
2090 if sb(x-1,y)=st(x-1,y)-1 then en=en
+10
2100 if sb(x,y-1)=st(x,y-1)-1 then en=en
+10
2110 next:next
2120 return
2200 rem * ende *
2210 print d$;"▒▒eine immerwaehrende exp
losion    "
2220 if pl=1 then print "sie gewannen";:
goto 2240
2230 print "ich gewann";
2240 print " in";nt;"▒ zuegen"
2250 input "(return druecken)";an$
2260 if left$(an$,1)="j" then poke646,14
:clr:goto10
2270 end
3000 rem * hauptprogramm *
3010 gosub 400:rem initialisieren
3020 if cm then 3040
3030 pl=1:gosub 800:rem zug des spielers
3040 pl=-1:gosub 1600:rem zug des comput
ers
3050 nt=nt+1
3060 goto 3030
```

New York, New York

Dieses originelle und aufregende Spiel versetzt Sie in einen Hubschrauber, hoch über New York, und Sie schauen auf die Straßen der Stadt hinunter. Noch sind Sie allein, aber bald wird Verkehr auftauchen und Sie müssen ihn regeln. Sie tun dies, indem Sie die Verkehrsampeln steuern. Wenn ein Auto (dargestellt durch ein Quadrat) eine rote Ampel erreicht, wird es anhalten. Jede Ampel hat ein Zeichen – drücken Sie die Taste mit dem Zeichen, und die Ampel wird grün und das Auto wird seine Reise fortsetzen. Am Anfang des Spieles haben Sie 1ØØ Punkte. Sie erzielen 1Ø Punkte, jedesmal wenn ein Auto eine Kreuzung überquert, und 25 Punkte für jedes Auto, das die Stadt durchquert. Aber jedesmal, wenn zwei Autos an einer Kreuzung zusammenstoßen, verlieren Sie 1ØØ Punkte und die Straße ist zeitweise gesperrt. Wenn Sie keine Punkte mehr haben, ist das Spiel zu Ende – wenn nicht, dauert es etwa fünf Minuten. Je mehr Punkte Sie bekommen, desto besser sind Sie als Verkehrspolizist!



```
10 rem * c-64 new york, new york *
15 poke 53281,7
20 ti$="000000":tb=ti:pu=100
30 print"▒▒▒▒▒▒▒▒▒ ] ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒ ] ▒▒▒▒▒▒▒▒";
40 print"▒▒▒▒▒▒▒▒▒a] ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
b] ▒▒▒▒▒▒▒▒";
50 print"          ] c       ┬ d         ┬ e
 ] f      ";
60 print"──────────[───────────[───────────[────────
─[──────────";
70 print"       g ]       h ]         i ]         j
 ]        ";
80 print"▒▒▒▒▒▒▒▒ ]k▒▒▒▒▒▒▒ ]l▒▒▒▒▒▒▒▒ ]m▒▒▒▒▒▒▒
 ]n▒▒▒▒▒▒▒▒";
```

```
90 a$="▒▒▒▒▒▒▒ ] ▒▒▒▒▒▒ ] ▒▒▒▒▒▒▒ ] ▒▒▒▒▒▒ ]
 ▒▒▒▒▒▒▒":print a$ a$ a$;
100 print"▒▒▒▒▒▒▒o] ▒▒▒▒▒▒p] ▒▒▒▒▒▒▒q] ▒▒▒▒▒
▒r] ▒▒▒▒▒▒▒";
110 print"▒▒▒▒▒▒▒ ] s     ] t      ] u
  ] ▒▒▒▒▒▒▒";
120 print"▒▒▒▒▒▒▒├[────────[────────[─────
──[┤▒▒▒▒▒▒▒";
130 print"▒▒▒▒▒▒▒ ]     v ]      w ]
x ] ▒▒▒▒▒▒▒";
140 print"▒▒▒▒▒▒▒ ]y▒▒▒▒▒▒ ]z▒▒▒▒▒▒▒ ]0▒▒▒▒▒
▒ ]1▒▒▒▒▒▒▒";
150 printa$ a$ a$;
160 print"▒▒▒▒▒▒▒2] ▒▒▒▒▒▒3] ▒▒▒▒▒▒▒4] ▒▒▒▒▒
▒5] ▒▒▒▒▒▒▒";
170 print"       ] 6     ] 7      ] 8
  ] 9     ";
180 print"────────[────────[────────[─────
──[────────";
190 print"     + ]     | ┴        ┴
* ]       ";
200 print"▒▒▒▒▒▒▒ ]~▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒ ]=▒▒▒▒▒▒▒";
210 print"▒▒▒▒▒▒▒ ] ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒ ] ▒▒▒▒▒▒▒";
300 print"▒▒▒▒▒▒▒ ] ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒ ] ▒▒▒▒▒▒▒▒";
400 aw=40:sc%=1023:cl=55295
500 dimx%(aw),y%(aw),r%(aw),l%(41),a%(3)
,f%(9)
510 fori=0to41:readl%(i):next
520 data 47,72,90,98,107,115,166,174,183
,191,209,217,226,234,367,375,384,392
530 data 410,418,427,494,503,511,529,537
,546,554,687,695,704,712
535 data 730,738,747,755
540 data 806,814,823,831,849,874
550 fori=0to41:a=peek(l%(i)+sc%):pokel%(
i)+sc%,aor128
560 pokel%(i)+cl,2:next
590 a%(0)=1:a%(1)=-1:a%(2)=40:a%(3)=-40
600 a=rnd(-ti):deffnr(x)=int(rnd(1)*x)
1000 fori=0toaw:r%=r%(i):ifr%=0then1099
1010 x%=x%(i):y%=y%(i)
1020 ifr%and1thenx%=x%+1:f%=98:o0%=226:o
%=64:s%=40:ifx%>40then1200
1030 ifr%and2thenx%=x%-1:f%=226:o0%=98:o
%=64:s%=-40:ifx%<1then1200
1040 ifr%and4theny%=y%+1:f%=97:o0%=225:o
%=93:s%=-1:ify%>23then1200
```

```
1050 ifr%and8theny%=y%-1:f%=225:o0%=97:o
%=93:s%=1:ify%<0then1200
1060 ifr%and16then1100
1070 ifr%and32then1400
1080 ifr%and64theno%=91:r%=r%-64:r%(i)=r
%:pu=pu+10
1090 goto1300
1099 next:goto2000
1100 if(peek(x%(i)+40*y%(i)+s%+cl)and15)
<>2thenr%(i)=(r%or32)andnot16:goto1300
1110 pu=pu-1:goto1099
1200 pu=pu+25:pn=pn+1:r%(i)=0:p%=-1:goto
1330
1300 p%=x%+40*y%:ps%=peek(p%+sc%):ifps%=
f%orps%=160thenpu=pu-1:goto1099
1310 a=peek(p%+s%+sc%):ifa<>32thenr%(i)=
r%or16
1320 ifps%=o0%thenf%=160
1325 v%=peek(x%(i)+40*y%(i)+sc%):ifv%=16
0theno%=o0%
1327 ifv%=102thenr%(i)=0:ao=ao+1:pu=pu-1
00:goto1099
1330 pokep%+sc%,f%:pokex%(i)+40*y%(i)+sc
%,o%:x%(i)=x%:y%(i)=y%:goto1099
1400 p%=x%+40*y%:a=peek(p%+sc%):ifa=f%th
en1099
1410 ifa<>91then1500
1420 vb=0:forj=0to3:a=2~j:b=peek(p%+sc%+
a%(j)):ifnot(b=64orb=93)thenvb=vbora
1430 next
1440 b=2~fnr(4):if(bandvb)=0thenr%(i)=bo
r64:goto1330
1450 ifvb=15then1099
1460 goto1440
1500 ao=ao+1:r%(i)=0:ifa<>102thenf%(af)=
p%:af=af+1:pu=pu-200
1510 f%=102:goto1330
2000 ifr%(tl)=0then2100
2010 tl=tl+1:iftl>awthentl=0
2020 goto3000
2100 a=fnr(4):r%(tl)=2~a:ona+1goto2110,2
120,2130,2140
2110 x%(tl)=1:goto2200
2120 x%(tl)=40:goto2200
2130 y%(tl)=0:goto2300
2140 y%(tl)=23:goto2300
2200 iffnr(2)=1theny%(tl)=3:goto3000
2210 y%(tl)=19:goto3000
2300 iffnr(2)=1thenx%(tl)=8:goto3000
2310 x%(tl)=33:goto3000
3000 geta$:ifa$=""then3090
```

```
3010 a=asc(a$):ifa>64anda<91thena=a-65:g
oto3100
3020 ifa>47anda<58thena=a-22:goto3100
3030 ifa=43thena=36:goto3100
3040 ifa=92thena=37:goto3100
3050 ifa=64thena=38:goto3100
3060 ifa=42thena=39:goto3100
3070 ifa=94thena=40:goto3100
3080 ifa=61thena=41:goto3100
3090 goto4000
3100 p%=l%(a):a=peek(p%+cl):if(aand15)=2
thena=5:goto3110
3105 a=2
3110 pokep%+cl,a
3120 goto3000
4000 rem * ausgabe *
4010 print""spc(
18)mid$(ti$,4,1)":"right$(ti$,2)""
4020 ifpu<0thenpu=0
4030 print""spc(18)right$("    "+str$(p
u),4)
4040 ifpu=0then9000
4100 iffnr(25)orf%(0)=0then4200
4110 pokef%(0)+sc%,91:fori=0to8:f%(i)=f%
(i+1):next:af=af-1
4200 ifti-tb>18e3then9000
5000 goto1000
9000 fori=0to3e3:next:print" noch ein
mal? (j/n)"
9010 geta$:ifa$="j"thenrun20
9020 ifa$="n"then9040
9030 goto9010
9040 if pu>0 then print"sie haben";pu;
" punkte gemacht"
9050 ifpu<0thenend
9060 print"          sie haben die fuenf
minuten"
9070 print"              nicht durchgehalte
n"
```

Schlüssel

Die Suche nach einem Schlüssel, den man verloren hat, kann bestenfalls eine ärgerliche Erfahrung sein, aber wenn man ihn so schnell wie möglich finden muß und er außerdem im Computer versteckt liegt, dann wird die Sache äußerst mühselig, aber auch sehr herausfordernd.

Der Computer hat intern folgenden Ausdruck aus 2∅ Einsen und Nullen gespeichert.

10100101011101010110

Diese Zahlen können zyklisch verschoben werden, das heißt, die Ziffern werden vom rechten Ende nach links geschoben. Dies wird dreimal gemacht und die Zahlen in jeder Spalte werden addiert, zum Beispiel:

```
Reihe 1  → 00101011101010110101
Reihe 2  → 01110101011010100101
Reihe 3  → 01001010111010101101
           --------------------
Summe    → 02212122223030311303
```

Dies alles bekommen Sie nicht zu sehen. Eigentlich ist dies der Schlüssel, den Sie finden müssen.

Was Ihnen gezeigt wird, sind die drei Zahlenreihen, jede wieder verschoben.

Zum Beispiel, wenn die oberste Reihe zwei Positionen, die mittlere Reihe vier Positionen und die unterste Reihe acht Positionen verschoben ist, sehen Sie

```
10101110101011010100
01010110101001010111
11101010110101001010
--------------------
20000112111123310122
```

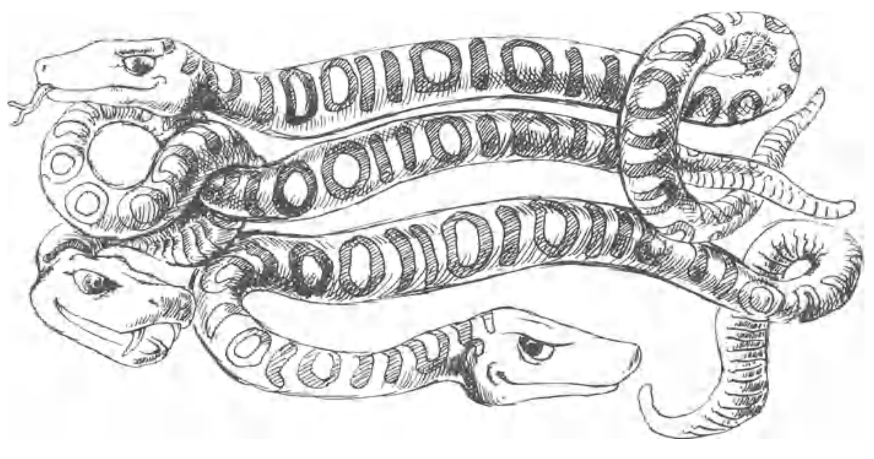
Ihnen wird nicht die Summe der neuen Spalten gezeigt, sondern die Differenz für jede Spalte zwischen dieser Summe und der ersten "versteckten" Summe.

In der ersten Spalte, beispielsweise, war die versteckte Summe ∅ und die neue Summe ist 2, deshalb sehen Sie 2; in der zweiten Spalte war die versteckte Summe 2 genau wie die neue Summe, daher sehen Sie ∅. Sie sollen jetzt die drei Reihen verschieben, bis sie identisch sind mit dem versteckten Schlüssel, dann sind natürlich alle Differenzen ∅. Die Reihen sind numeriert (die oberste Reihe ist 1, die mittlere 2, und die unterste 3), so daß, wenn Sie eingeben

REIHE = 2

SCHRITTE = 1

die mittlere Reihe eine Position nach links geschoben wird. Wie "wenige" Drehungen brauchen Sie um den Schlüssel zu finden? Es wurde bereits in 1Ø geschafft.



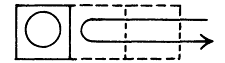
```
10 rem * c-64 schluessel *
20 a$="10100101011101010110"
30 dim t(3,19)
40 goto 1000
100 rem * initialisieren *
110 for k=0 to 19
120 t(0,k)=val(mid$(a$,k+1,1))
130 t(1,k)=t(0,k)
140 t(2,k)=t(0,k)
150 next k
160 gosub 400
170 for k=0 to 19
180 t(3,k)=t(0,k)+t(1,k)+t(2,k)
190 next k
200 gosub 400
210 return
300 rem * reihe verschieben *
310 for i=1 to s
320 h=t(r,0)
330 for k=0 to 18
340 t(r,k)=t(r,k+1)
350 next k
360 t(r,19)=h
370 next i
380 return
400 rem * zufaelliges verschieben *
410 for r=0 to 2
420 s=int(20*rnd(0))
430 gosub 300
440 next r
```

```
450 return
500 rem * bildschirm *
510 print "♥"
520 for r=0 to 2
530 print "          ";
540 for k=0 to 19
550 print chr$(t(r,k)+48);
560 next k
570 print
580 next r
590 print "          ────────────────────
"
600 f=0
610 print "          ";
620 for k=0 to 19
630 d=abs(t(0,k)+t(1,k)+t(2,k)-t(3,k))
640 print chr$(d+48);
650 f=f or d
660 next k
670 print
680 return
800 rem * eingabe *
810 print
820 print spc(9) "reihe      =";
825 input r
830 print spc(9) "schritte   =";
835 input s
840 r=r-1
850 return
1000 rem * haupt *
1010 gosub 100:rem initialisieren
1020 nt=0
1030 gosub 500:rem bildschirm
1040 if f=0 then 1090
1050 gosub 800:rem eingabe
1060 gosub 300:rem verschieben
1070 nt=nt+1
1080 goto 1030
1090 print
1100 print spc(10) "geschafft in";nt;" d
rehungen"
```

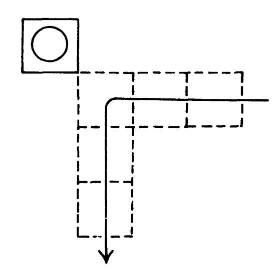
Der Schwarze Kasten

Nur mit einem Laser ausgerüstet, sollen Sie die Standorte einer handvoll Atome, die in einem gewaltigen Schwarzen Kasten versteckt sind, entdecken. Er besteht aus 8 mal 8 Würfeln. Atome können in jedem der Würfel, außer in denen der äußersten Schicht, versteckt liegen. Im Kasten sind immer fünf Atome. Diese Atome werden Laserstrahlen reflektieren oder ablenken nach den folgenden Regeln:

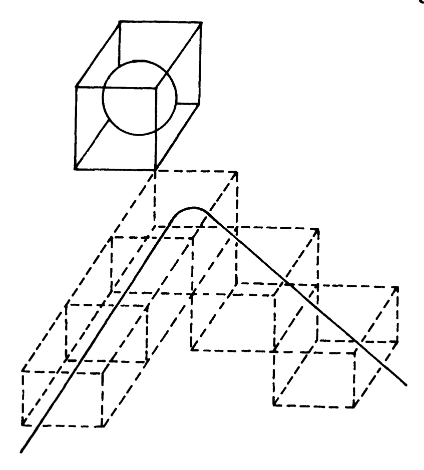
- ein Strahl, der ein Atom trifft, wird gerade zurückreflektiert.



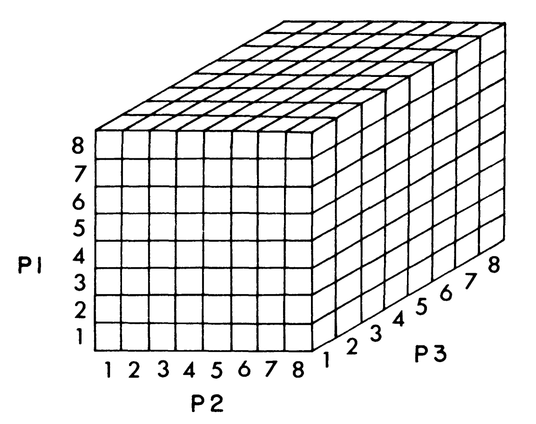
- ein Strahl, der durch einen direkten Nachbarwürfel geht, wird rechtwinklig reflektiert.



- ein Strahl, der durch einen diagonal anliegenden Nachbarwürfel eines Atoms geht, wird in eine Richtung reflektiert, die erhalten werden kann durch Addition zweier rechtwinkliger Reflektionen.



Der Kasten ist wie folgt numeriert:



Der Computer wird Sie nach P1, P2 und P3 fragen; das sind die Koordinaten der Position, wo der Strahl in den Kasten eintreten soll. Natürlich muß zumindest eine der Koordinaten 1 oder 8 sein, da der Strahl ja nur an einer Oberfläche eintreten kann. Wenn Sie den Schuß eingegeben haben, wird der Computer Ihnen sagen, wo der Strahl ausgetreten ist.

Denken Sie daran, der Strahl kann durch mehr als ein Atom reflektiert worden sein.

Nachdem Sie ein paar Schüsse abgegeben haben, sollten Sie imstande sein, herauszufinden, wo die Atome sind – aber wieviel sind denn "ein paar"?

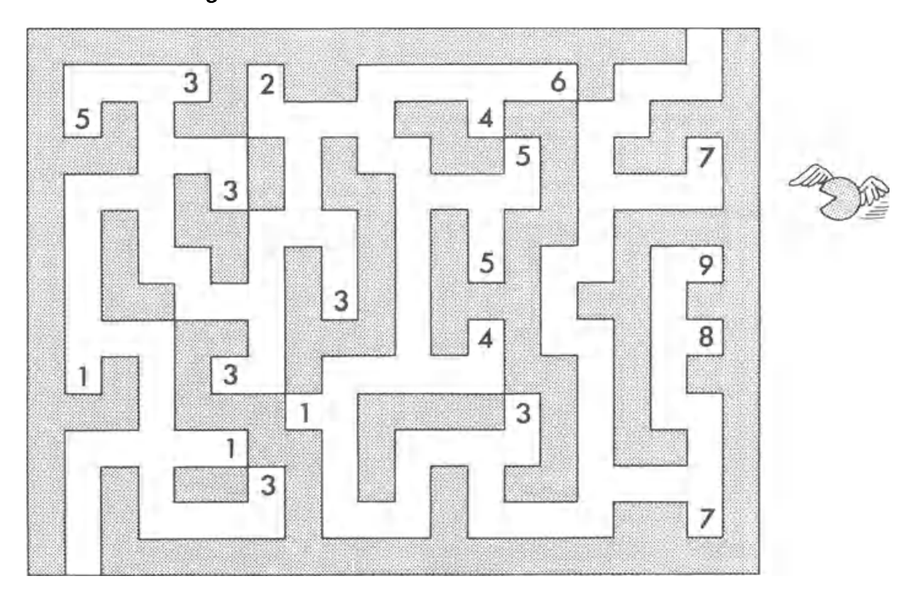
```
10 rem * c-64 der schwarze kasten *
20 dim b(9,9,9)
30 goto 1000
100 rem * eingabe des schusses *
110 print "geben sie den schuss ein:"
120 gosub 500:rem eingabe
130 dx=(px=8)-(px=1)
140 dy=(py=8)-(py=1)
150 dz=(pz=8)-(pz=1)
160 if dx=0 and dy=0 and dz=0 then 120
170 return
200 rem * ergebnis berechnen *
210 for tx=-1 to 1
220 for ty=-1 to 1
230 for tz=-1 to 1
240 if b(px+tx,py+ty,pz+tz)<>1 then 280
250 dx=dx-tx
260 dy=dy-ty
270 dz=dz-tz
280 next tz,ty,tx
290 dx=sgn(dx):dy=sgn(dy):dz=sgn(dz)
300 px=px+dx:py=py+dy:pz=pz+dz
```

```
310 if (px=0)+(px=9)+(py=0)+(py=9)+(pz=0
)+(pz=9)=0 then 210
320 print "ergebnis:";px-dx;py-dy;pz-dz
330 return
400 rem * eingabe vermutung *
410 print "vermutung eingeben"
420 gosub 500:rem eingabe
430 if b(px,py,pz)=1 then print "richtig
!":goto 450
440 print "falsch!"
450 return
500 rem * eingabe der koordinaten *
510 print "koordinaten eingeben:"
520 input "  p1=";px
530 if px<1 or px>8 then 520
540 input "  p2=";py
550 if py<1 or py>8 then 540
560 input "  p3=";pz
570 if pz<1 or pz>8 then 560
580 return
1000 rem * hauptprogramm *
1005 print "⊠"
1010 for at=1 to 5
1020 px=6*rnd(0)+2
1021 py=6*rnd(0)+2
1022 pz=6*rnd(0)+2
1023 if b(px,py,pz)=1 then 1020
1025 b(px,py,px)=1
1030 next
1040 gosub 100:rem schusseingabe
1050 gosub 200:rem resultat berechnen
1060 print "schiessen oder raten? (s/r)"
1070 get in$
1080 if in$="s" then 1040
1090 if in$<>"r" then 1070
1100 gosub 400:rem vermutung eingeben
1110 goto 1060
```

Schatzsuche

Haben Sie jemals davon geträumt, auf die Suche nach versteckten Schätzen zu gehen? Durch wilde und feindliche Gegenden zu reisen, auf dem Land zu leben und im Freien zu schlafen, bis Sie das ferne verbotene Land, wo der funkelnde Schatz versteckt ist, erreichen. Wenn dem so ist, ist dies genau das richtige Spiel für Sie.

Sie werden hoch in die Rocky Mountains versetzt und erhalten eine Karte, die Ihnen zeigt, wo die Schätze, die Sie suchen, versteckt sind und was sie wert sind. Um sie zu erreichen, müssen Sie sich einen Weg bahnen entlang sich windender enger Pfade – ein falscher Schritt bedeutet den sicheren Tod –, indem Sie die folgenden Tasten verwenden.



O, für einen Schritt nach oben

K, für einen Schritt nach links

:, für einen Schritt nach rechts

., für einen Schritt nach unten

Sie entdecken die Schätze ganz einfach, indem Sie nämlich drauftreten.

Aber warten Sie . . . so einfach ist es nicht. Sie haben doch nicht etwa im Ernst geglaubt, daß es so einfach wäre? Die Schätze, die Sie suchen, sind scharf bewacht durch ein paar entsetzliche und äußerst bösartige Vögel, die Sie attackieren werden, wenn Sie nicht vorsichtig sind. Wenn die Vögel Sie erblicken, werden Sie vor Furcht erstarren.

Das einzige, was Sie tun können, um sie zu stoppen, ist die <SPACE> Taste drücken.

Wenn Sie sich der Herausforderung immer noch stellen wollen, starten Sie Ihre Suche links unten auf der Karte. Sammeln Sie so viele Schätze wie Sie können, und bringen Sie sie weg, durch den Ausgang rechts oben. Das gibt Bonuspunkte und eine weitere Chance, den Pfaden entlang zu gehen und Schätze einzusammeln. Sie können also so viele Schatzsuchläufe machen, wie Sie schaffen.

Unglückerweise scheuchen Sie jedesmal, wenn Sie durchgehen, mehr von diesen schrecklichen Vögeln auf, die zunehmend aggressiver werden.

Das Risiko ist groß, aber die Belohnung auch, also fassen Sie sich ein Herz und los geht's!

```
10 rem * c-64 schatzsuche *
15 poke 53281,1
20 sc=1024
30 d$="■■■■■■■■■■■■■■■■■■"
40 goto 1000
100 rem * bildschirm *
110 print "■";
120 print "▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒ ▒ "
130 print "▒   3▒2▒▒     6▒   ▒ "
140 print "▒5▒ ▒▒    ▒▒4▒▒  ▒▒▒ "
150 print "▒▒▒   ▒ ▒  ▒▒5▒ ▒▒7▒ "
160 print "▒   ▒3▒ ▒▒    ▒    ▒ "
170 print "▒ ▒ ▒▒   ▒ ▒ ▒▒ ▒▒▒▒ "
180 print "▒ ▒  ▒ ▒ ▒ ▒5▒  ▒ 9▒ "
190 print "▒ ▒▒   ▒3▒ ▒▒▒ ▒▒ ▒▒ "
200 print "▒   ▒▒ ▒▒▒ ▒4▒  ▒ 8▒ "
210 print "▒1▒ ▒3 ▒     ▒▒ ▒ ▒▒ "
220 print "▒▒▒ ▒▒▒1 ▒▒▒▒3▒ ▒  ▒ "
230 print "▒    1▒▒ ▒    ▒ ▒▒ ▒ "
240 print "▒ ▒ ▒▒3▒ ▒ ▒ ▒▒    ▒ "
250 print "▒ ▒    ▒   ▒    ▒▒7▒ "
260 print "▒ ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒ "
300 y=14:x=1
310 pl=40*y+x
320 poke sc+pl,0
330 print d$;p2,nc;
340 return
400 rem * spieler-bewegung *
410 t1=400-int(30*nc*rnd(0))
420 for du=1 to t1
430 get di$:if di$="" then 510
440 vx=(di$="k")-(di$=":")
450 vy=(di$="o")-(di$=".")
460 np=40*(y+vy)+(x+vx)
470 pk=peek(sc+np):if pk<>32 and (pk<49
or pk>57) then 510
480 poke sc+pl,32:x=x+vx:y=y+vy:pl=np
490 po=pk-48:if po<0 then 510
```

```
500 p1=p1+po:p2=p2+po:print d$;p2;
510 poke sc+p1,0:if p1=40 then du=t1
520 next du:if p1>40*15 then 400
530 return
600 rem * vogelangriffe *
610 dt=15-3*nc
615 for i=1 to 10:get di$:next
620 bu=40*y+20
630 pk=peek(sc+bu)
640 poke sc+bu,31
650 for de=1 to dt:next
660 get di$
670 if di$=" " or pk=0 then return
680 poke sc+bu,pk
690 bu=bu-1
700 goto 630
1000 rem * hauptprogramm *
1010 gosub 100:rem bildschirm
1020 gosub 400:rem bewegen
1030 if p1<40 then nc=nc+1:p2=p2+nc*p1:p
1=0:goto 1010
1040 gosub 600:rem vogel
1050 if pk<>0 then poke sc+bu,pk:p1=p1+1
:goto 1020
```



I.T. – Das Abenteuer des Jahrhunderts

In diesem verrückten Abenteuer werden Sie in das Erdinnere reisen und dort einem äußerst fremdartigen Wesen begegnen. Sein Name ist I. T.; das steht für Intra Terrestrial.

Wenn Sie noch nie vorher ein Adventure-Spiel gespielt haben, machen Sie sich keine Sorgen über Spielregeln – es gibt keine! Sie müssen alles selbst herausfinden, während Sie unter der Erdoberfläche reisen, außergewöhnliche Werke vollbringen und dabei schrecklichen Gefahren ins Auge sehen.

Wenn es Ihnen zuviel wird, geben Sie <HILFE> ein: Schauen Sie, womit der Computer Ihnen dann unter die Arme greift.

So, tippen Sie das Spiel ein, schließen Sie eine gute Versicherung ab, geben Sie den <RUN>-Befehl und.ab gehts!





```
10 rem * c-64 i.t. *
20 gosub 5000
30 mv=8:nr=0:goto 50
40 gosub 200
50 gosub 500
60 gosub 400
70 if nw<6 then 40
100 print "■glueckwunsch! ihnen gelang e
s, sich selbst umzubringen!"
110 end
200 rem * eingabe *
210 nr=0:mv=0
220 input "was jetzt";c$
230 for i=1 to 14
240 if left$(m$(i),len(c$))=c$ then mv=i
:i=14
250 next i
260 if mv>6 then goto 310
270 if mv=0 then print "das verstehe ich
 nicht":goto 200
280 nr=r(rm,mv):if nr<>0 then 310
290 print "dieser weg ist versperrt"
300 goto 200
310 print "■";
320 return
400 rem * ausgabe *
```

```
410 print "sie sind i";:print d$(rm)
420 if rm>=25 or (lo and t(1)=0 or t(1)=
rm) then 430
425 print "hier ist es finster":goto 440
430 for i=1 to 12
432 if t(i)=rm then print "ein ";t$(i);"
 ist hier"
434 next i
435 if lo and rl>120 then print "ihre la
mpe ist ausgegangen":lo=0
436 if lo and rl>100 then print "die lam
pen-batterien werden schwaecher"
440 for i=1 to 6
450 if r(rm,i)<>0 then print "sie koenne
n gehen nach ";m$(i)
460 next i
470 print tab(8);"***"
480 return
500 rem * bearbeitungen *
510 if nr<>0 then rm=nr
520 th=0
530 for i=1 to 12
540 if t(i)=rm then th=i:i=12
550 next i
560 hl=0
570 for i=1 to 6
580 if h(i,1)=rm then hl=i:i=6
590 next i
600 rem * allgemeine bearbeitungen *
610 if rm<12 and rr<>8 then gosub 4200
620 rl=rl-lo
630 rem * persoenliche bearbeitungen *
640 if mv<7 then 660
650 on mv-6 gosub 1000,1200,1300,1400,16
00,1700,1800,1900
660 rem * handlers *
670 on hl gosub 2000,2300,2600,2700,2900
,3100
680 return
1000 rem * mord *
1010 if not(hl=0 or hl>4)then 1020
1015 print"zeit ist das einzige, was sie
 hier totschlagen koennen!":return
1020 tp=7:gosub 4100
1030 ifpo=0thenprint"mit blossen haenden
 angreifen? das ist selbstmord!":return
1040 print "sie versuchen ";:print h$(hl
);" zu ermorden"
1050 print "sie schwingen ihr schwert,";
1055 if (hl=1 or hl=2) and (h(hl,2)=6 or
 h(hl,2)=5) then 1070
```

```
1060 if rnd(0)<.5 then print " aber es b
ewegt sich schnell weg...":goto 1090
1070 print " und sie versetzen ihm einen
 fuerchterlichen hieb !"
1080 h(hl,3)=h(hl,3)-1
1090 if rnd(0)>.2 then 1120
1100 print "dies ist ihre chance!! sie k
oennen es   nochmal treffen, bevor es";
1105 print" wieder zu sich kommt."
1110 input "moechten sie (j/n)";an$:prin
t "▒";
1115 if an$="j" then print "sie reissen
ihr schwert hoch, ";:goto 1055
1120 if h(hl,3)>1 then return
1130 if h(hl,3)=1 then print "das opfer
ist schwer verwundet":return
1140 print "sie haben ihr opfer umgebrac
ht."
1150 h(hl,1)=0:t(hl+8)=rm:hl=0
1160 return
1200 rem * hilfe *
1210 print "▒moegliche befehle:▒"
1220 for i=1 to 14:print m$(i):next
1230 print "▒druecken sie eine taste"
1240 get in$:if in$="" then 1240
1250 print "▒";
1260 return
1300 rem * aufheben *
1305 if (t(1)=0 or t(1)=rm) and lo or rm
>24 then 1310
1306 print "sie sehen nicht einmal die h
and vor den  augen, geschweige denn ";
1307 print"etwas zum aufheben!":return
1310 if th=0 then print "hier gibt es ni
chts aufzuheben":return
1320 print "ich nehme ";t$(th)
1330 p(lp)=th:lp=lp+1:t(th)=0:th=0
1340 return
1400 rem * hinlegen *
1410 if lp=1 then print"sie haben nichts
!":return
1420 print "was hinlegen":input d$:print
 "▒";
1430 tp=0:for i=1 to 12
1440 if d$=left$(t$(i),len(d$)) then tp=
i:i=12
1450 next i
1460 if tp=0 then print "das verstehe ic
h nicht":return
1470 gosub 4100
1480 if po=0 then print "sie tragen kein
```

```
 ";:print t$(tp):goto 1520
1490 print "ich lege ";t$(tp);" hin."
1500 th=p(po):t(th)=rm:lp=lp-1:p(po)=p(l
p)
1520 input "moechten sie mehr hinlegen";
an$
1530 print "[illegible]";
1540 if an$="j" then 1420
1550 return
1600 rem * ausruestung *
1610 if lp=1 then print "sie haben gar n
ichts!":goto 1640
1620 print "sie haben dabei:"
1630 for i=1 to lp-1:print t$(p(i)):next
1640 print "sie koennen";5-nw;" weitere
wunden ertragen"
1650 return
1700 rem * bandage *
1710 tp=6:gosub 4100
1720 if po=0 then print "sie schaffen es
 nicht ohne bandage":return
1730 print "es schaut nicht sehr gut aus
. ich brauche alles, was sie haben."
1750 print "ok, das haelt fuer eine weil
e."
1760 nw=0:lp=lp-1:p(po)=p(lp)
1770 return
1800 rem * lampe ein/aus *
1810 if t(1)>0 and t(1)<>rm then print "
erst eine lampe haben!":return
1820 if rl>120 then print "ihre batterie
n sind leer":return
1830 lo=not lo
1840 print "die lampe ist jetzt ";
1850 if lo then print "ein":return
1860 print "aus"
1870 return
1900 rem * lesen *
1910 tp=4:gosub 4100
1920 if po<>0 then 1930
1925 print "hier gibt es nichts zu lesen
!":return
1930 if not lo and rm<25 then print "es
ist zu dunkel zum lesen":return
1940 t$="dr fhas hm chdrdl atbg":gosub 4
000
1950 t$="dr kztsds:":gosub4000
1952 print
1955 t$="mhll cdm bdqadqtr tmc":gosub 40
00
1960 t$="dhm jnqmatdmcdk.":gosub4000
```

```
1970 t$="kdf dr hmr edtdq.":gosub4000
1980 print "das ist alles!"
1990 return
2000 rem * i.t. *
2010 if th=3 then h(1,2)=3:t(th)=0:th=0
2020 on h(1,2) goto 2030,2060,2090,2190,
2220
2030 print tab(5);"** i.t. **"
2034 print "i.t. ist hier, er spricht ei
ndringlich, aber sie koennen nichts ";
2035 print"verstehen.":print
2038 print
2050 h(1,2)=2:return
2060 t$="h.s. ekdgs tl vzrrdq. dq rbgzts
 udgyvhedks ztr.":gosub 4000
2075 print
2080 return
2090 t$="h.s. sqhmjs vhd dhm":gosub 4000
2095 t$="udqctdqrsdmcdq.":gosub 4000
2100 t$="mzbg dhmdq vdhkd":gosub 4000
2110 t$="ezdmfs dq vhdcdq zm yt":gosub 4
000
2115 t$="qdcdm tmc rhd jndmmdm":gosub400
0
2120 t$="hgm okndsykhbg udqrsdgdm !":gos
ub 4000
2133 t$="fdgdm rhd qtmsdq tmc":gosub 400
0
2138 t$="ehmcdm rhd chd nldfz-mtkk-anlad
 !":gosub4000
2140 t$="dhm hqqdq oqnedrrnq lndbgsd chd
":gosub4000
2145 t$="vdks ydqrsndqdm!":gosub4000
2150 t$="hbg gzad lhs cdl cqzbgdm, cdq c
hd anlad":gosub4000
2155 t$="advzbgs, fdjzdloes":gosub4000
2160 t$="hbg ahm dqkdchfs! rhd ltdrrdm c
hd dqcd  qdssdm !":gosub4000
2170 print "ploetzlich bricht i.t. zusam
men!"
2180 h(1,2)=4:return
2190 print "i.t. ist anscheinend bewusst
los, sie    muessen es jetzt alleine ";
2200 print"schaffen...":print"viel gluec
k!":print
2210 h(1,2)=5:return
2220 print "i.t. ist hier. er ist bewuss
tlos.":print
2240 return
2300 rem * drache *
2310 if th=8 then h(2,2)=5:t(th)=0:th=0
```

```
2320 on h(2,2) goto 2330,2370,2400,2430,
2470,2520
2330 print "hier ist ein gigantisches mo
nster. es   rollt die augen und bruellt"
2335 print:print "bist' ein keks?"
2360 h(2,2)=2:return
2370 print "das monster bruellt lauter u
nd lauter:▣"
2380 print "bist' ein keks ?!!"
2390 h(2,2)=3:return
2400 print "das monster bruellt weiter,
und es wird ziemlich aggressiv."
2420 h(2,2)=4:return
2430 print"das monster verpasst ihnen ei
nen furchtbaren hieb.ihr kopf wackelt!"
2450 nw=nw+1
2460 h(2,2)=fnrd(3)+1:return
2470 print "das monster sagt ueberrascht
: kekse ?!"
2480 print "und faengt an zu fressen. ei
n enormes"
2495 print "   ! ! b a n g ! !"
2497 print "und es faellt in schlaf."
2500 r(16,6)=1
2510 h(2,2)=6:return
2520 print "das monster schlaeft."
2530 return
2600 rem * schlange *
2610 print "hier ist eine schlaengelnde
schlange"
2620 if rnd(0)<.4 then return
2630 if lp=1 or rnd(0)<.5 then 2670
2640 rn=fnrd(lp-1):t(p(rn))=12+fnrd(12):
lp=lp-1:p(rn)=p(lp)
2650 print "mit einer schnellen bewegung
 schnappt   sie etwas und"
2670 print "schlaengelt weg"
2680 h(3,1)=h(3,1)+3:if h(3,1)>24 then h
(3,1)=h(3,1)-8
2690 return
2700 rem * hoellenhund *
2710 on h(4,2) goto 2720,2750,2780
2720 print "hier ist ein riesiger hoelle
nhund."
2730 print"er schaut sie an, als ob er s
ie zum"
2735 print"fruehstueck verspeisen moecht
e."
2740 h(4,2)=2:return
2750 print "cerberus greift an und beiss
t sie       gefaehrlich."
```

```
2760 nw=nw+1
2770 h(4,2)=3:return
2780 print "cerberus knurrt und macht si
ch bereit"
2790 print"fuer einen weiteren angriff."
2800 h(4,2)=1+fnrd(2):return
2900 rem * bombe *
2910 print "die omega-null-bombe ist hie
r. sie       sehen drei draehte";
2920 print " zwischen der bombe und dem
zuender:":print
2925 print "einen gruenen     (g), ";
2927 print "einen gelben      (y), ";
2929 print "und einen roten   (r)":print
2930 print "sie muessen zwei loesen, um
sie zu       entschaerfen."
2940 input "welcher zuerst  ";x$
2950 input "und welchen jetzt";y$
2960 cb=0:c$="gyr":for i=1 to 3
2970 cb=cb-(x$=mid$(c$,i,1))-(y$=mid$(c$
,i,1))
2980 next i
2990 if cb<2 then print "▒vorsicht!":pri
nt "falsche eingabe!":goto 2910
3000 if asc(x$)*asc(y$)=5822 then 3030
3005 for i=1 to 10:poke 53281,1
3010 print "▒ * grosse  explosion *▒"
3012 print "      ! wolkenpilze !▒"
3016 print "sie waren tapfer, aber es wa
r ihnen"
3020 print "nicht vergoennt..."
3022 for j=1 to 4:poke 53281,9:next j,i
3025 end
3030 print "▒▒▒▒▒▒▒▒▒▒▒▒*** glueckwunsch
***"
3040 print "▒ihnen gelang, was keinem ge
lang"
3060 end
3100 rem * grill *
3130 if t(2)=rm and t(12)=rm then th=8:t
(th)=rm:t(2)=0:t(12)=0:h(6,2)=2
3140 on h(6,2) goto 3150,3180,3220
3150 print "hier ist ein riesiger grill,
 mit "
3165 print "einem grossen feuer darunter
."
3170 return
3180 print " ein greller blitz erleuchte
t den ort."
3185 for i=1 to 40: poke 53281,1:next i
3186 poke 53281,2
```

```
3187 print "und ein penetranter geruch s
teigt in ihre nase"
3210 h(6,2)=3:return
3220 print "jetzt ist alles ruhig, sogar
 der        penetrante geruch ist weg."
3250 h(6,2)=1:return
4000 rem * text dekodieren *
4010 tn$="":for i1=1 to len(t$):c$=mid$(
t$,i1,1)
4020 if c$="z" then c$="a":goto 4030
4025 if c$>="a" and c$<"z" then c$=chr$(
asc(c$)+1)
4030 tn$=tn$+c$
4040 next:if not in then print tn$
4050 return
4100 rem * was sie dabeihaben *
4110 po=0:for i=1 to lp-1
4120 if p(i)=tp then po=i:i=lp-1
4130 next i
4140 return
4200 rem * raetsel *
4210 rt=rt+1
4220 if rm=s(rt) then rr=rr+1
4230 if rt<8 then return
4240 if rr=8 then 4310
4250 for q=1 to 80:poke 53281,0:next q
4255 print "totale dunkelheit umgibt sie
 und sie"
4265 print "fuehlen, dass sie emporgehob
en werden."
4270 print "einen moment lang sind sie b
ewusstlos."
4280 poke 53281,2
4290 rt=1:rr=1:rm=1
4300 return
4310 print "sie hoeren ein eigenartiges
geraeusch,  wie wenn etwas";
4330 print "weggeschoben wird."
4340 for dl=1 to 3000:next dl
4350 print "jetzt ist es vorbei."
4360 r(2,1)=3
4370 return
5000 rem * initialisieren *
5010 print "▒▓"spc(7) "einen moment gedu
ld, bitte."
5020 dim r(36,6),d$(36),m$(14),h(6,3),h$
(6),t(12),t$(12),p(12),d(8)
5030 def fnrd(x)=int(x*rnd(0)+1)
5100 rem * verbindungen *
5110 for i=1 to 36
5120 r(i,1)=i+1:r(i,2)=i-1:r(i,3)=i+4:r(
```

```
i,4)=i-4
5130 next i
5140 for i=0 to 24 step 12
5150 for j=1 to 9 step 4:r(i+j+3,1)=0:r(
i+j,2)=0:next
5160 for j=1 to 4:r(i+j+8,3)=0:r(i+j,4)=
0:next
5170 next i
5180 r(1,5)=16:r(7,5)=15:r(32,6)=13:r(13
,5)=32:r(35,6)=18:r(18,5)=35
5190 for i=1 to 15:read rm:read mv:r(rm,
mv)=0:next
5200 rem * initialisieren der parameter
*
5210 rm=36:po=1:lp=1:rl=0:lo=0:nw=0:rt=0
:rr=0
5300 rem * arrays fuellen *
5310 in=-1
5320 for i=1 to 14:read m$(i):next
5330 for i=1 to 12:read t$:gosub4000:t$(
i)=tn$:next
5340 for i=1 to 6:read t$:gosub 4000:h$(
i)=tn$:next
5350 for i=1 to 36:read t$:gosub 4000:d$
(i)=tn$:next
5360 for i=1 to 12:read t(i):next
5370 for i=1 to 6:read h(i,1),h(i,3):h(i
,2)=1:next
5380 for i=1 to 8:read s(i):next
5390 in=0
5400 return
5500 data 21,1,22,2,22,1,23,2,18,1,19,2,
16,3,20,4,11,1,12,2,7,1,8,2,7,4,3,3,2,1
5510 data osten,westen,norden,sueden,obe
n,unten,toeten,hilfe,aufheben,hinlegen
5515 data ausruestung,bandage,lampe,lese
n
5520 data kzlod,jnqmatdmcdk,vzrrdqadtsdk
,jnbgatbg,akzss,azmczfd
5524 data rbgvdqs,jdjr,jndqodq  unm h.s.
5525 data qhdrdmkdhbgd,gzkad rbgzmfd,sns
dq gtmc
5530 data h.s.,lnmrsdq,rbgzmfd,bdqadqtr,
fqhkk,anlad
5540 data m cdq rszqs-gndgkd,m cdq e-gnd
gkd,l fdgdhlfzmf,l jnmsqnkkqztl
5541 data m cdq f-gndgkd,m cdq t-gndgkd,
m cdq z-gndgkd,l rbgvzqydm rzzk
5545 data m cdq d-gndgkd,m cdq g-gndgkd,
m cdq kddq-gndgkd,m cdq kddqd
5550 data m cdq jkdhmdm gndgkd,m cdq edk
```



```
rdmgndgkd,m cdq rshmjdq-gndgkd
5551 data m cdq cqzbgdmgndgkd,m cdq rbgk
zmfdmgndgkd,m cdq fdkadm gndgkd
5555 data m cdq edtbgsdm gndgkd,"Ilzm dhm
dl rshmjdmcdm nqs"
5556 data m cdq drrfqnssd,m cdq dmcd-gnd
gkd
5557 data m cdq ezqa-fqnssd,m cdq dhr-gn
dgkd,m dhmdq khbgstmf
5560 data l vzkc,l.vzkc,l vzkc,l vzkc,l
vzkc
5561 data l vzkc,l vzkc,l vzkc,l vzkc,l
vzkc,l vzkc
5570 data 34,30,28,21,14,15,13,0,0,0,0,0
5580 data 34,2,16,15,17,4,29,2,8,1,25,1
5590 data 1,5,9,10,11,7,6,2
```

Der Wolf und die fünf Geißlein

Ein Märchen der Gebrüder Grimm? Nein, ein mitreißendes Brettspiel. Dieses Spiel wird auf einem auf dem Bildschirm gezeigten Damebrett gespielt.

Wenn das Spiel beginnt, sind die Ziegen auf der unteren Bretthälfte verteilt und durch kleine Quadrate mit einer Zahl dargestellt. Der Wolf befindet sich oben links im Spielfeld. Los geht's – Sie sind der Wolf und gewinnen, wenn Sie drei Ziegen verspeisen, und der Computer gewinnt, wenn eine der Ziegen Sie verspeist!

(Diese wölfeverspeisenden Ziegen sind verblüffende Viecher!)

Sie dürfen anfangen. Auf der unteren Schirmhälfte sehen Sie folgendes:

SIE DUERFEN . . . ZUEGE EINGEBEN

beispielsweise

SIE DUERFEN 2 ZUEGE EINGEBEN

Die Zahl der Züge liegt immer zwischen 1 und 3, und sie sagt Ihnen, wieviele Züge Sie machen dürfen, während Sie dran sind.

Der Wolf – mit anderen Worten Sie – darf vertikal oder horizontal ziehen, aber nie diagonal. Jeder Zug muß mit den Cursor-Tasten eingegeben werden.

Wenn der letzte Zug einer Serie Sie auf ein Feld mit einer Ziege bringt, gehört sie Ihnen. Sie dürfen nie ein Feld überqueren, das eine Ziege enthält.

Es gibt insgesamt 5 Ziegen, und diese können einander überspringen. Ziegen können nur in eine Richtung ziehen. Die Zahl der Schritte, die sie machen dürfen, ist auf den Ziegen selbst gezeigt, und liegt immer zwischen 1 und 5. Zum Beispiel, wenn eine Ziege die Zahl 3 trägt, kann sie wie folgt ziehen:

3 Schritte nach links, oder

3 Schritte nach rechts, oder

3 Schritte vorwärts, oder

3 Schritte rückwärts

Sie kann nicht 1 Schritt vorwärts und 2 Schritte nach links ziehen. Wenn Sie (der Wolf) gezogen haben, zeigt der Computer

JETZT BIN ICH DRAN

Wenn Sie dran sind, zeigt der Computer

SIE DUERFEN . . . ZUEGE EINGEBEN

Am Ende des Spiels sagt der Computer entweder

SIE GEWINNEN

oder, schlimmer

SIE HABEN VERLOREN



```
10 rem * c-64 der wolf und die fuenf gei
sslein *
20 dim b(10,10),r(5),c(5),nm(5)
30 c$(0)="▒▒"
40 c$(1)="  "
50 d$="▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣"
60 goto 2000
1000 rem * ein feld zeichnen *
1010 on ca+2 goto 1020,1030,1040
1020 r=rw:c=cw:tx$="ww":goto 1070
1030 tx$=c$(r+c and 1):goto 1070
1040 r=r(gt):c=c(gt)
1050 tx$="g"+right$(str$(nm(gt)),1)
1060 ca=gt
1070 print left$(d$,3+r);tab(2*c-1);spc(
7) tx$
1080 b(r,c)=ca
1090 return
1100 rem * initialisieren *
1110 rem schachbrett zeichnen
1120 print "▣";
1130 for r=1 to 10
1140 for c=1 to 10
1150 ca=0:gosub 1000
1160 next c,r
1170 rem wolf hinstellen
1180 rw=1:cw=1:nm=int(rnd(0)*3+1)
1190 ca=-1:gosub 1000
1200 rem geisslein hinstellen
1210 for gt=1 to 5
```

```
1220 r(gt)=int(rnd(0)*10+1)
1230 c(gt)=int(rnd(0)*10+1)
1240 nm(gt)=int(rnd(0)*5+1)
1250 if b(r(gt),c(gt))<>0 then 1220
1260 ca=1:gosub 1000
1270 next gt
1280 return
1300 rem * wolf bewegen *
1310 print d$;"sie duerfen";nm;"▮ zuege
eingeben"
1320 for du=1 to nm
1330 get mv$:if mv$="" then 1330
1350 rd=(mv$="▣" and rw>1)-(mv$="▣" and
rw<10)
1360 cd=(mv$="▮" and cw>1)-(mv$="▮" and
cw<10)
1370 te=b(rw+rd,cw+cd)
1380 if te<0 or te>0 and du<nm then 1330
1390 nm(te)=0
1400 sc=sc+sgn(te)
1410 r=rw:c=cw
1420 ca=0:gosub 1000
1430 rw=rw+rd:cw=cw+cd
1440 ca=-1:gosub 1000
1450 next du
1460 nm=int(rnd(0)*3+1)
1470 return
1500 rem * zug des computers *
1510 print d$;"jetzt bin ich dran
    "
1520 rem bewertung
1530 mx=-400
1540 for gt=1 to 5
1550 if nm(gt)=0 then 1600
1560 cg=c(gt):rg=r(gt)+nm(gt):gosub 1800
1570 rg=r(gt)-nm(gt):gosub 1800
1580 rg=r(gt):cg=c(gt)+nm(gt):gosub 1800
1590 cg=c(gt)-nm(gt):gosub 1800
1600 next gt
1610 rem geisslein ziehen
1620 c=c(gb):r=r(gb)
1630 ca=0:gosub 1000
1640 r(gb)=rb:c(gb)=cb
1650 nm(gb)=int(rnd(0)*5+1)
1660 gt=gb
1670 ca=1:gosub 1000
1680 if mx<400 then return
1690 print d$;" sie haben verloren
     "
1700 end
1800 rem * status *
```

```
1810 if abs(r(gt)-rw)+abs(c(gt)-cw)=nm t
hen sq=200:goto 1820
1815 sq=0
1820 if rg<1 or rg>10 or cg<1 or cg>10 t
hen return
1830 if b(rg,cg)>0 then return
1840 di=abs(rw-rg)+abs(cw-cg)
1850 if di=0 then sq=500:goto 1930
1860 if di=nm or di=nm-2 then return
1870 if di<=5 then sq=sq+35
1880 sq=sq-40*((rg=rw)+(cg=cw))
1890 for g=1 to 5
1900 sq=sq-10*((rg<>r(g))+(cg<>c(g)))
1910 next
1920 if sq+rnd(0)<mx then 1940
1930 mx=sq:gb=gt:rb=rg:cb=cg
1940 return
2000 rem * hauptprogramm *
2010 gosub 1100:rem initialisieren
2020 gosub 1300:rem wolf ziehen
2030 if sc=3 then 2060
2040 gosub 1500:rem zug des computers
2050 goto 2020
2060 print d$;"   sie gewinnen         !
        "
2070 end
```

Alphabahn

Dieses Programm demonstriert wiederum die Nützlichkeit der Sprites. Nachdem Sie RUN eingetippt haben, erscheint eine befahrene Autobahn mit vielen Wagen. Ihre Aufgabe ist nun, sicher über diese Straße zu gelangen. Damit aber nicht genug: Wenn Sie genau hinsehen, entdecken Sie ein komplettes Alphabet über der Autobahn, A, B, C, und so weiter. Sie sehen schon, Ihre Aufgabe wird immer schwieriger . . .

Bei Spielbeginn fragt der Computer nach dem Spielniveau, welches dann die Geschwindigkeit der Wagen beeinflußt. Jedoch fahren nicht alle Wagen gleich schnell, was das Ganze nochmals erschwert.

Falls Sie das Pech haben, in einen Unfall zu geraten, fragt der Computer:

NOCHMALS?

Durch J werden Sie wundersam geheilt, und das Spiel beginnt von neuem.

```
10 rem * c-64 alphabahn *
20 print "■"
30 v= 53248: rem * niedrigstes sprite-re
gister
40 poke v+21, 0
50 av= peek( v+30): rem kollisions-regis
ter initialisieren
60 poke 2040, 14
70 poke v+39, 1
80 rem niveau-eingabe
90 print "niveau (1-5) ?"
100 get l$
110 if l$> "5" or l$< "1" goto 100
120 l= val( l$): print "■"
130 rem sprite 0 definieren
140 for i=0 to 62: read q
150 poke 832+i, q: next i
160 for i= 0 to 62: poke 896+i, 0: next
i: rem  sprite 0 legen
170 poke 53281, 6: poke 53280, 6: rem fa
rbe
180 rem bildschirm
190 print "■■     strassenrennen   "
200 print "■           niveau:";l
210 print "■■■  abcdefghijklmnopqrstuvwx
yz"
220 for i= 1 to 14
230 print "■
  "
240 next i
250 print "■■■■■■■■■■■■■■■■■■■■■■";
```

```
260 rem  sprites 1-7 initialisieren
270 for i= 1 to 7
280 poke 2040+ i, 13
290 poke v+2*i, 240
300 poke v+2* i+1, 93+ 16*i
310 poke v+ 39+i, 2*i-1
320 next i
330 poke v+42, 2
340 poke v+21, 254
350 for i= 1 to 26: rem hauptprogramm
360 poke 56334, peek( 56334) and 254: re
m zeichen setzten
370 poke 1, peek( 1) and 251
380 for j= 0 to 7
390 poke 896+ 3*j, peek( 53248+ i*8+j)
400 next j
410 poke 1, peek( 1) or 4
420 poke 56334, peek( 56334) or 1
430 poke v, 32+ i*8
440 poke v+1, 98
450 poke v+21, 255
460 for av=1 to 250: next av
470 print " ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒";
480 rem spielen
490 for k= 1 to 7
500 jo=peek(56321)
510 if not jo and 1 and peek( v+1)> 98 t
hen poke v+1, peek( v+1)- 2
520 if not jo and 2 then poke v+1, peek(
 v+1)+ 2
530 x= peek( v+2*k)- 5*l* rnd( 1)-10- i/
2
540 if x<0 then x= x+240
550 poke v+2*k, x
560 if peek( v+30) and 1 then print: pri
nt "unfall !!!": goto 690
570 if peek( v+1)= 218 goto 600
580 next k
590 goto 490
600 for av= 1 to 250: next av
610 rem naechster buchstabe
620 print chr$( 64+i); "▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
;
630 poke v+21, 254
640 next i
650 rem  end
660 poke v+21, 0: print "▒gratuliere!"
670 for av= 1 to 2000: next av
680 if l<5 then l= l+1: goto 190
690 for av= 1 to 1000: next av
700 getav$:if av$<> "" goto 700
```

```
710 print "nochmals? (j/n) ";
720 get a$
730 if a$= "" goto 720
740 if a$= "j" then run
750 if a$<> "n" then 720
760 poke v+21, 0
770 end
780 data 0, 0, 0, 0, 15, 254, 0, 15, 254
, 0, 15, 254, 0, 127, 254
790 data 0, 159, 254, 0, 159, 254, 0, 25
5, 254, 0, 240, 0, 0, 255, 254
800 data 0, 124, 124, 0, 56, 56, 0, 0, 0
, 0, 0, 0, 0, 0, 0
810 data 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0, 0, 0, 0, 0, 0, 0
```

Auf dem Markt

Sind Sie schon jemals durch einen Markt gegangen und gestaunt über die Geschwindigkeit, mit der ein Verkäufer eine Preisliste addieren kann? Spielen Sie dieses Spiel mit Ihrer Familie und finden Sie heraus, was für eine Figur Sie als Markthändler abgeben würden.

Sie werden Zahlenpaare von zunehmender Länge sehen, die Sie addieren sollen. Sie werden schnell herausfinden, daß es nicht so einfach ist, wie es aussieht. Um zu sehen warum, betrachten Sie folgende Summe

$$\begin{array}{r} 75856 \\ +\ 37637 \end{array}$$

Normalerweise würden Sie erst die Ziffern in der rechten Spalte zusammenzählen, dann die nächste Spalte, und so weiter. Jedoch mit dem Computer müssen Sie die Lösung mit der linken Spalte beginnend eingeben. Dieser Unterschied macht das Spiel so vertrackt.

Wieviele Zahlen können Sie richtig in der gegebenen Zeit von etwa einer Minute addieren? Können Sie unseren Rekord von acht brechen?

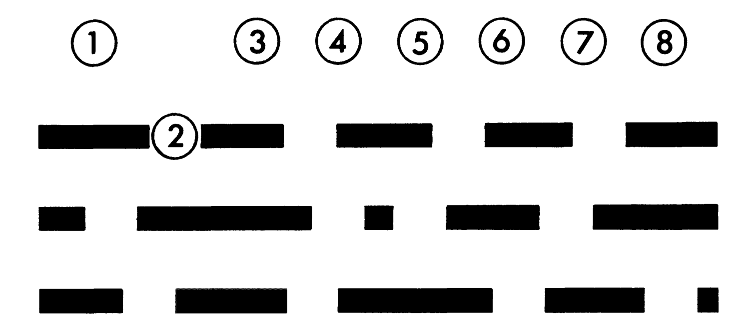


```
10 rem * c-64 auf dem markt *
20 dim jo$(9)
30 for i=0 to 9
40 read jo$(i)
50 next
60 print "[illegible]";
70 nr=0:ti$="000000":mx=1
80 goto 1000
90 data dummkopf,knirps,abc-schuetze,ang
estellter,autohaendler,makler
```

```
100 dataerfolgreicher boersenfachmann,to
pmanager,millionaer,lehrer
200 rem * zufaelliges addieren *
210 n1$=str$(int(1+(mx-1)*rnd(0)))
220 n2$=str$(int(1+(mx-1)*rnd(0)))
230 an=val(n1$)+val(n2$)
240 print tab(nr+12-len(n1$));n1$
250 print tab(nr+12-len(n2$));n2$;"+"
255 print"[illegible]";
260 for du=0 to nr+1
270 print  "-";
280 next
285 print
290 return
300 rem * eingabe antwort *
310 input "ergebnis";an$
320 return
500 rem * pruefen der antwort *
505 print:print
510 if val(an$)=an then print "richtig![illegible]
":nr=nr+1:goto 540
520 print "falsch!"
525 print
530 print "es war:";an
535 print "zeit:";str$(int(ti/.6)/100);"
sec"
540 return
1000 rem * hauptprogramm *
1010 mx=mx*10
1020 gosub 200:rem addieren
1030 gosub 300:rem antwort eingeben
1040 gosub 500:rem check
1050 if val(an$)=an then 1010
1060 print "job:";:print jo$(nr)
```

Fallout

Am Anfang dieses einfachen aber fesselnden Spieles werden Sie acht horizontale Balken mit Löchern sehen. Über den Balken sind acht Steine, die durch die Löcher fallen können. In diesem Beispiel



ist Stein 2 bereits in ein Loch gefallen. Der Sinn dieses Spieles liegt darin, alle acht Steine durch die Balken zu bekommen. Dies können Sie erreichen, indem Sie die Balken so verschieben, daß die Löcher eine Linie bilden und so die Steine durchfallen können.

Um einen bestimmten Balken zu bewegen, geben Sie einen Befehl in der folgenden Form ein

BRS

wobei

B die Nummer des Balkens ist (die Balken sind von 1 bis 8 von oben nach unten durchnumeriert).

R ist die Richtung, in die Sie verschieben wollen (l für links, r für rechts), und

S ist die Zahl der Schritte, um die der Balken verschoben werden soll.

Zum Beispiel

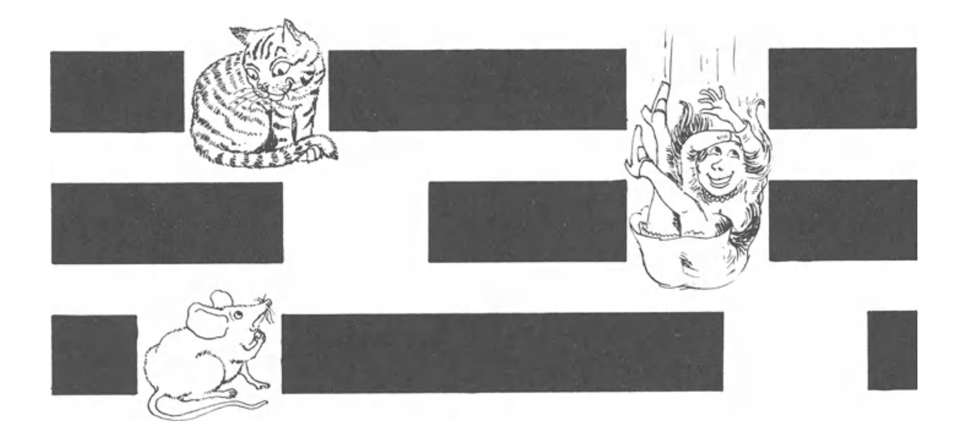
3R2

verschiebt Balken 3 um 2 Schritte nach rechts. Dies klingt vielleicht wie ein sehr simples Spiel, aber wenn Sie wirklich mit Spielen anfangen, werden Sie herausfinden, daß es eine ziemlich harte Nuß ist.

```
10 rem * c-64 fallout *
20 dim h(16,8)
30 print "[illegible]";
40 rem * mit balken fuellen *
50 for x=0 to 16
60 for y=1 to 8
70 h(x,y)=1
80 next y,x
90 rem * 4 zufallsloecher *
100 for y=1 to 8
110 for b=1 to 4
120 x=int(rnd(0)*16+1)
130 h(x,y)=0
140 next b,y
150 rem * 1-8 obenhin drucken *
160 for b=1 to 8
170 h(2*b-1,0)=b+100
180 next b
190 gosub 610:rem * zeichnen
200 gosub 910:rem * aendern
210 gosub 610:rem * zeichnen
300 rem * befehl holen *
310 input z$
320 y=val(left$(z$,1))
330 if y<1 or y>8 then 510:rem * loesche
n
340 d$=mid$(z$,2,1)
350 dx=val(mid$(z$,3))
360 if dx=0 then 510:rem * loeschen
370 if d$<>"l" and d$<>"r" then 510:rem
* loeschen
380 if d$="l" then gosub 1110:rem * link
s schieben
390 if d$="r" then gosub 1210:rem * rech
ts schieben
400 br=br+1
410 gosub 910:rem * aendern
420 gosub 610:rem * zeichnen
430 if sc>=36 then end:rem * wenn alle s
teine durch sind
440 if dx>1 then dx=dx-1:goto 380:rem *
dx-mal wiederholen
500 rem * loeschen
510 print "[illegible]"
520 for y=10 to 21
530 print "                    "
540 next y
550 print "                    ";
560 print "[illegible]"
570 goto 310:rem * befehl holen
600 rem * zeichnen
```

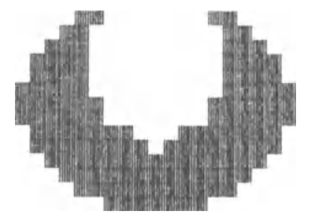
```
610 print "[illegible]";
620 for y1=0 to 8
630 rem * balkennummer drucken (oder nic
ht)
640 if y1=0 then print "     ";
650 if y1>0 then print spc(8) str$(y1);"
> ";
660 for x=0 to 16
670 z=h(x,y1)
680 gosub 810:rem * element drucken
690 next x
700 print
710 next y1
720 print spc(8) "punktzahl";sc;" schrit
te";br
730 return
800 rem * element drucken
810 if z=0 then print " ";
820 if z=1 then print "[illegible]";
830 if z>100 then print  chr$(z-52);
840 return
900 rem * aendern
910 for y1=8 to 0 step -1:rem * suche vo
n unten nach oben
920 for x=0 to 16
930 ty=y1
940 if h(x,y1)<100 then 1020
950 if ty>=8 then 1060:rem * unten angek
ommen
960 if h(x,ty+1)<>0 then 1020:rem * bloc
kiert
970 rem * stein durch
980 h(x,ty+1)=h(x,ty)
990 h(x,ty)=0
1000 ty=ty+1
1010 goto 950:rem * weiterschieben
1020 next x
1030 next y1
1040 return
1050 rem * unten angekommen
1060 sc=sc+h(x,ty)-100
1070 h(x,ty)=0
1080 goto 1020
1100 rem * balken nach links verschieben
*
1110 tm=h(0,y)
1120 for x=0 to 15
1130 h(x,y)=h(x+1,y)
1140 next x
1150 h(16,y)=tm
1160 return
```

```
1200 rem * balken nach rechts verschiebe
n *
1210 tm=h(16,y)
1220 for x=16 to 1 step -1
1230 h(x,y)=h(x-1,y)
1240 next x
1250 h(0,y)=tm
1260 return
```



UFO Angriff

Beschreiben wir das Spiel zuerst so realistisch wie möglich. In der unteren Hälfte des Schirmes befindet sich eine Figur wie diese:



Kleine Quadrate fallen vom oberen Bildschirmrand hinunter. Sie müssen die Figur mit Hilfe der Cursor-Kontrolltasten bewegen, damit die Quadrate sie nicht mehr treffen.

Wem das zu dumm klingt, wie wäre es hiermit . . .

Sie sind Kommandant eines der besten Raumschiffe im Universum. Wenn Sie majestätisch durch die Milchstraße kreuzen, können Sie sich eines Gefühls des Stolzes auf dieses herrliche Gefährt unter Ihrem Befehl nicht erwehren. Dann, gerade wenn Sie sich der Basis nähern und das Ende der Reise in Sicht ist, schlägt das Schicksal zu –

UFO ANGRIFF!

Wie können Sie diesen unfreundlichen Eindringlingen von jenseits der Milchstraße entkommen?

Warum nicht Ihren

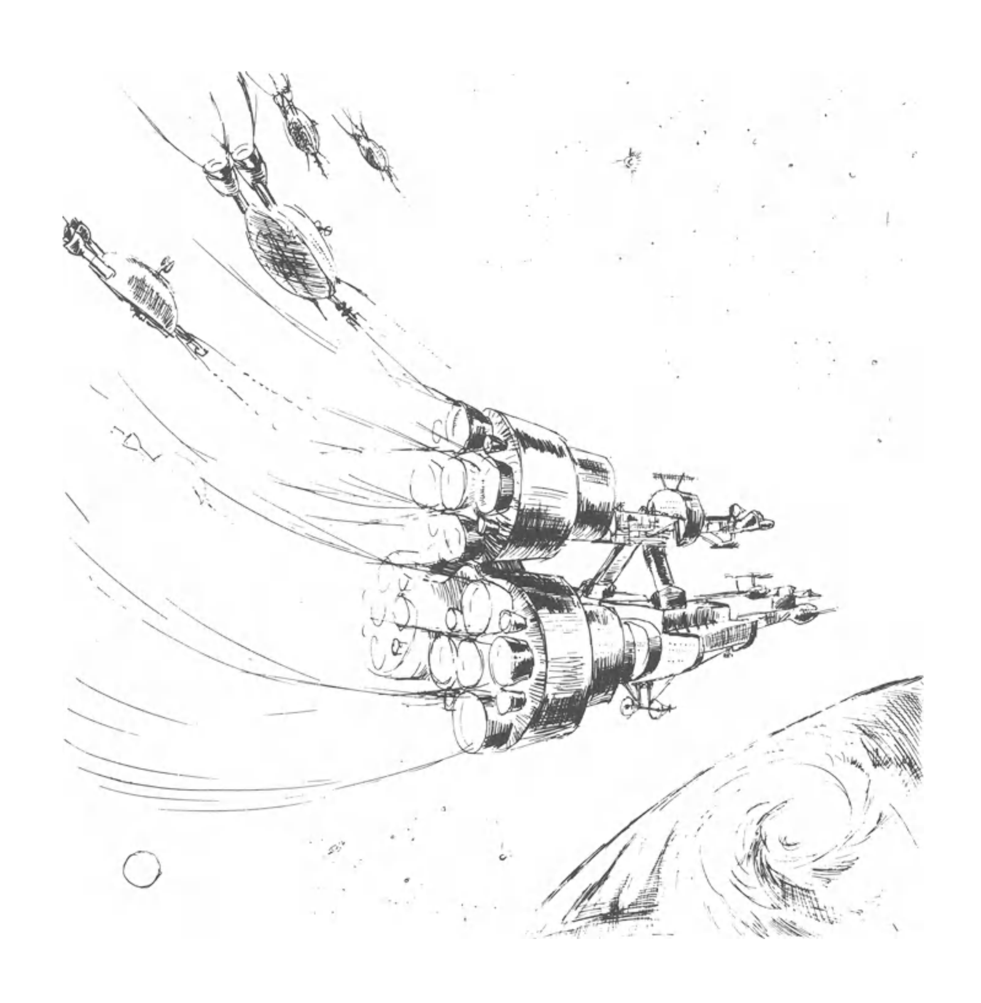
SUPERATOMAREN FLUCHTMOTOR

ausprobieren?

Je mehr UFOs Sie entkommen, desto mehr Punkte bekommen Sie.

Das klingt besser, oder? Wie dem auch sei, das Spiel ist schnell, einfach und macht viel Spaß beim Spielen. Und nur das zählt.

```
10 rem * c-64 ufo angriff *
20 sr=53248
30 poke2040,13
40 for i=0 to 62
50 read x
60 poke 13*64+i,x
70 next
80 poke sr+23,1:poke sr+29,1
```

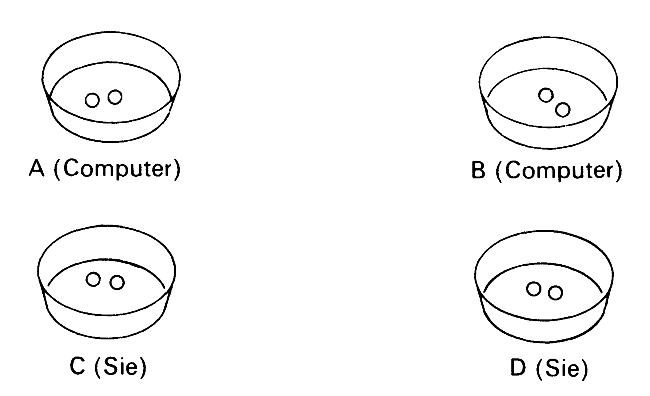


```
90 poke sr+1,112:poke sr+39,7:du=peek(sr
+31)
100 m$=" [illegible] [illegible]"
110 poke 646,1
120 input "[illegible]niveau (1-3)";le
130 if le<1 or le>3 then 120
140 print "[illegible]";:pl=21
145 pokesr+21,1
150 po=po+1
160 rem * ufo erzeugen *
170 sa=40*9+pl-40*int(4*rnd(0)+5-le)+3-6
*rnd(0)
180 get a$
```

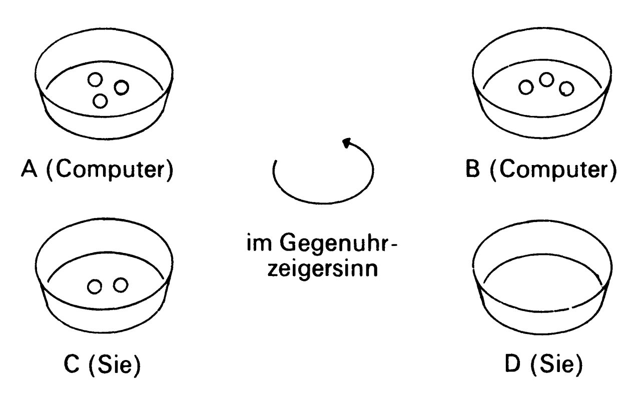
```
190 pl=pl+(a$="▌▌" and pl>0)-(a$="▌▌" and
pl<40)
200 rem * schiff und ufo bewegen *
210 p1%=pl*8
220 p2%=p1%/255:p1%=p1%-p2%*255
230 poke sr,p1%:poke sr+16,p2%
240 a1=0:a2=sa
250 if sa>255 then a1=255:a2=sa-255
260 print "▌";tab(a1);spc(a2);m$;
270 if peek(sr+31)and 1 <>0 then 310
280 sa=sa+40:if sa<10*40+pl then 180
290 print "▌▌ ";
300 if sa>=10*40+pl then 150
310 rem * explosion *
320 poke54272,0:poke54273,20
330 poke54277,127:poke54278,250
340 poke54276,129
350 for du=1 to 15
360 poke 54296,16-du
370 poke 53280,int(16*rnd(0))
380 poke 53281,int(16*rnd(0))
390 poke 36877,128+int(20*rnd(0))
400 for t=1 to 50:next t
410 next du
420 poke 54296,0
430 poke 53280,14:poke53281,6
440 poke54276,0
450 print "▌sie haben";po-1;"punkte"
460 rem sprite daten
470 data  0,0,0,0,0,0,0
480 data  0,0,0,0,0,12,6
490 data  0,24,3,0,56,3,128
500 data  120,3,192,120,3,192,248
510 data  3,224,252,7,224,252,7
520 data  224,127,31,192,127,191,192
530 data  63,255,128,31,255,0,15
540 data  254,0,3,248,0,0,0
550 data  0,0,0,0,0,0,0
```

Mini-Mancala

Mini-Mancala beruht auf einem alten arabischen Spiel. Es wird gespielt mit Steinchen, die zwischen Schälchen hin- und herbewegt werden. Es gibt vier Schälchen: A und B gehören dem Computer, C und D gehören Ihnen. Am Anfang des Spieles sind in jedem Schälchen zwei Steinchen.



Abwechselnd nehmen die Spieler die Steinchen von einem ihrer Schälchen und verteilen sie entgegen dem Uhrzeigersinn auf die anderen drei Schälchen. Beispielsweise könnten Sie die Steinchen aus Schälchen D wie folgt verteilen:



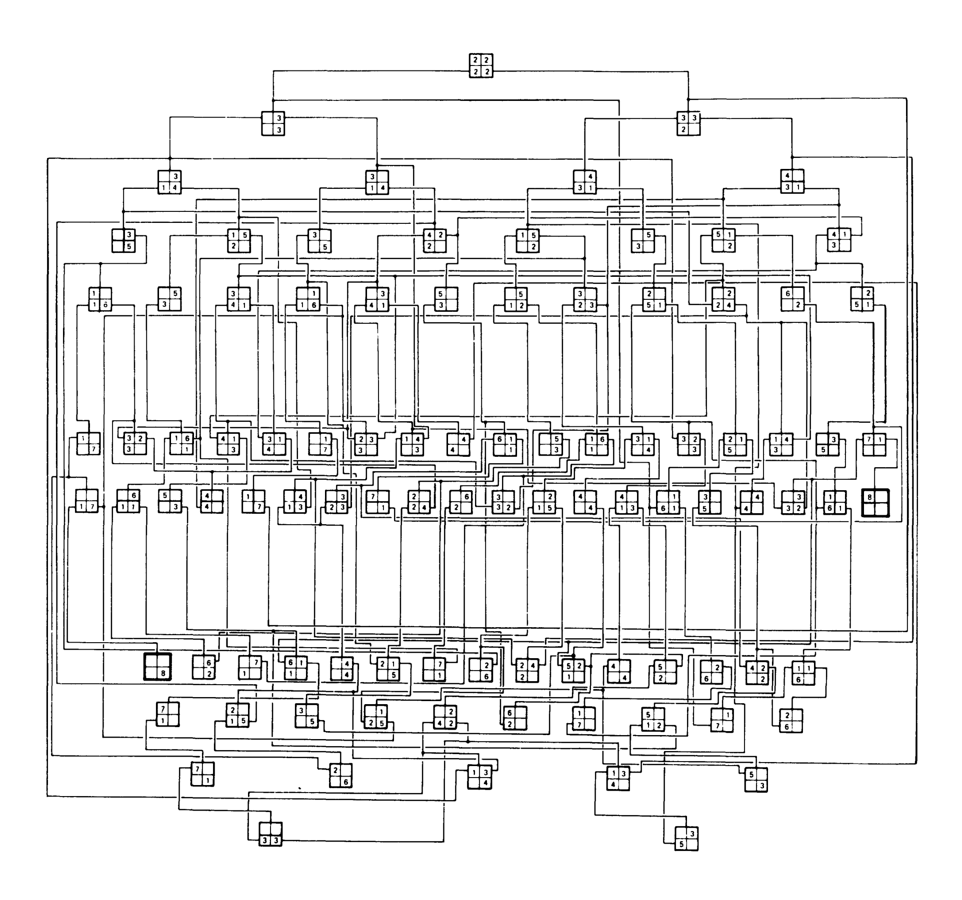
Nach diesem Zug ist Schälchen D leer. Es wird immer zumindest ein Schälchen leer sein, da während eines Zuges keine Steinchen in das Schälchen, von dem sie weggenommen wurden, zurückgelegt werden dürfen.

Wenn Sie alle Steinchen in Ihren eigenen Schälchen haben, dann haben Sie gewonnen.

Auf dem Computer werden die Schälchen dargestellt durch Quadrate mit Ziffern, die angeben, wieviel Steinchen sie enthalten. Sie können den Schwierigkeitsgrad wählen: 1, 2 oder 3 (1 ist am einfachsten), und auch, wer den ersten Zug hat. Der Computer wird Sie fragen, von welchen Ihrer Quadrate Sie die Steinchen wegnehmen wollen und wird Ihnen sagen, welcher Zug er selber macht. Die Verteilung der Steinchen auf dem Brett wird nach jedem Zug angegeben.

Sie werden vielleicht überrascht sein, daß dieses komplizierte Spiel in einem so kurzen Programm beschrieben werden kann. Das Diagramm zeigt alle möglichen Züge und ist ein excellentes Beispiel dafür, wie ein strategisches Spiel schematisch wiedergegeben werden kann.

Mini-Mancala ist eine Schöpfung von C. Freeling.



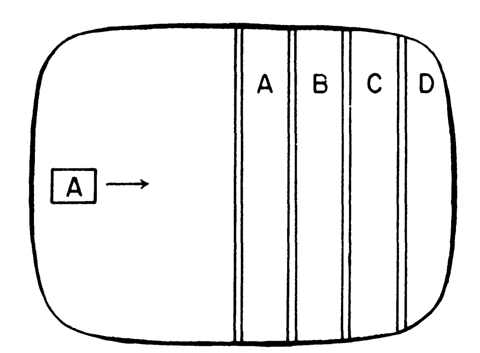
```
10 rem * c-64 mini-mancala *
20 dim b(3),su(3)
30 def fnsp(x)=15+80*(x and 2)+6*(x and
1)
40 for i=0 to 3
50 b(i)=2
60 read su(i)
70 next
80 data 2,0,3,1
90 e$="[illegible]"
110 goto 1000
200 rem * schirm zeichnen *
210 print"[illegible]" spc(13) "mini-mancala"
220 print
230 print spc(11) " a"spc(13)"b"
240 print spc(11) "   ┌─────┬─────┐"
250 print spc(11) "   ]     ]     ]"
260 print spc(11) "   ]  2  ]  2  ]"
270 print spc(11) "   ]     ]     ]"
280 print spc(11) "   ├─────[─────┤"
290 print spc(11) "   ]     ]     ]"
300 print spc(11) "   ]  2  ]  2  ]"
310 print spc(11) "   ]     ]     ]"
320 print spc(11) "   └─────┴─────┘"
330 print spc(11) " c"spc(13)"d"
340 return
400 rem * zug eingeben *
410 print e$"
              ";e$;
430 input"schaelchen (c,d)";s$
440 if s$="" then 410
450 s=-2*(s$="c")-3*(s$="d")
460 if s=0 or b(s)=0 then 410
470 return
500 rem * zug des computers *
510 print e$"jetzt bin ich dran
          "
520 for de=1 to 500:next de
530 if le<3 and rnd(0)*le<.6 then s=int(
rnd(0)*2):goto 560
540 q=b(2)+10*(b(1)+10*b(0))
550 s=1
560 if q=143 or q=134 or q=611 or q=116
then s=0
570 if b(s)=0 then s=1-s
580 print e$"ich nehme aus          "mi
d$("ab",s+1,1)
600 return
700 rem * im gegenuhrzeigersinn verteile
n *
710 d=s
```

```
720 for c=1 to b(s)
730 for de=1 to 400:next de
740 d=su(d)
750 if d=s then 740
760 b(s)=b(s)-1
770 print "■■■■■■■"spc(fnsp(s));b(s)
780 b(d)=b(d)+1
790 print "■■■■■■■"spc(fnsp(d));b(d)
800 next c
810 return
1000 rem * hauptprogramm *
1010 gosub 200:rem schirm zeichnen
1020 print e$"    niveau (1-3)";
1030 input le
1040 if le<1 or le>3 then 1020
1050 print  e$"moechten sie anfangen (j/
n)";
1060 input an$
1070 if an$="n" then 1120
1080 if an$<>"j" then 1050
1090 gosub 400:rem eingabe des zuges
1100 gosub 700:rem im gegenuhrzeigersinn
 verteilen
1110 if b(3)=8 then print e$"sie gewinne
n!                      ":goto 1160
1120 gosub 500:rem zug des computers
1130 gosub 700:rem im gegenuhrzeigersinn
 verteilen
1140 if b(0)=8 then print e$"ich gewinne
!                       ":goto 1160
1150 goto 1090
1160 end
```

Stop it!

Obwohl es nur ein kurzes Programm ist, gibt es einem schnellen und spannenden Spiel Gestalt.

Der Schirm schaut etwa so aus:



Das Quadrat links trägt einen Buchstaben. Wenn es über den Schirm bewegt wird, sollen Sie versuchen, es durch Drücken der <SPACE>-Taste zu stoppen, und zwar im Bereich mit dem gleichen Buchstaben.

Am Anfang des Spieles fragt der Computer nach dem Schwierigkeitsgrad (1 ist am einfachsten, 2 ist schwieriger, und 3 ist am allerschwierigsten) und auch, wie oft Sie spielen möchten.

Am Ende wird Ihnen Ihre Trefferzahl gezeigt, beispielsweise

Ø TREFFER AUS 1Ø

Macht nichts, mehr Glück beim nächstenmal!

```
10 rem * c-64 stop it! *
20 print "▣"
30 d$="■■■■■■■■■■■■■■■■■■■■■■■■■"
40 input "schwierigkeitsgrad (1-3)";di
50 dt=10-3*di
60 input "wie oft moechten sie spielen";
nt
70 print "▣"
80 for ro=0 to 24
90 print tab(31)"] ] ] ]"
100 next ro
110 print "▣";tab(31);" a b c d"
120 tu=tu+1
130 co=int(rnd(0)*4+1)
140 bu$=mid$("abcd",co,1)
150 ro=2*int(rnd(0)*12+1)
```

```
160 print "■";left$(d$,ro+1);
170 print " ";bu$;
180 for de=1 to dt*9:next
190 print "■■ ■■";
200 get a$
210 if pos(0)<39 and a$="" then 170
220 if pos(0)<>30+2*co or pos(0)=42 then
 get du$:goto 250
230 hi=hi+1
240 print "■■";hi;
250 if tu>=nt then print "■";hi;" treffe
r aus";tu:end
260 for a=1 to 500:next
270 goto 120
```

BAAB

Ein "thema con variatione" ist eine Melodie die, obwohl sie auf eine bereits gehörte aufbaut, doch ihre eigene Stimmung und Identität hat. Dieses Spiel könnte als ein "thema con variatione" betrachtet werden, da es einige Gemeinsamkeiten mit einem anderen Spiel in diesem Buch hat, aber dennoch seinen eigenen Reiz birgt.

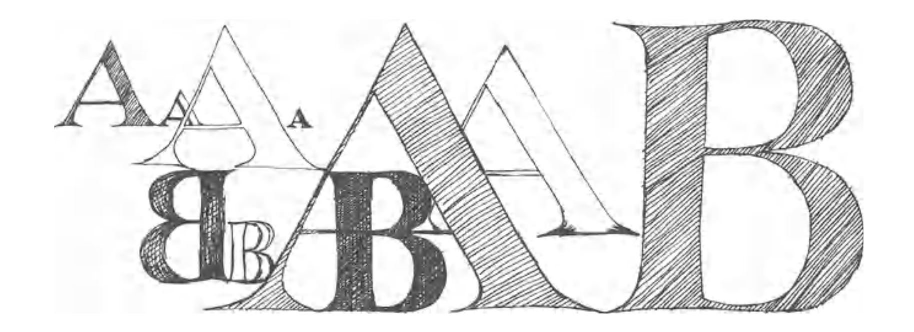
Sie werden 16 mit einer zufälligen Anordnung von As und Bs belegte Felder sehen, z. B.

B	A	B	A
A	A	A	B
A	B	A	B
B	B	A	B

Wenn Sie eins der Felder eingeben (unter Verwendung der auf dem Schirm gezeigten Taste) werden alle Buchstaben auf der horizontalen und vertikalen Reihe durch das Feld verändert, und zwar so, daß alle As zu Bs werden und umgekehrt. Ihr Ziel ist es, mit einem Schirm aufzuhören, der so ausschaut:

A	B	B	A
A	B	B	A
A	B	B	A
A	B	B	A

```
10 rem * c-64 baab *
20 dim b(4,4)
30 rem * initialisieren
40 for r=1 to 4
50 for c=1 to 4
60 b(r,c)=int(2*rnd(0)-1)
70 next c,r
80 print "■";
1000 rem * bildschirm zeichnen
1010 print "■■■";tab(16);"1234"
1020 for r=1 to 4
1030 print
1040 print tab(13);str$(r)+" ";
1050 for c=1 to 4
1060 if b(r,c) then print "b";:goto 1080
1070 print "a";
1080 next c,r
1100 rem * zug holen *
1110 print:print
1120 print spc(14) "reihe   =";
1125 input r
1130 if r<1 or r>4 then 1120
1140 for c=1 to 4
1150 b(r,c)=not b(r,c)
1160 next
1170 print spc(14) "spalte=";
1175 input c
1180 if c<1 or c>4 then 1170
1190 b(r,c)=not b(r,c)
1200 for r=1 to 4
1210 b(r,c)=not b(r,c)
1220 next
1230 goto 1000
```



Vokale und Konsonanten

Dieses Wettbewerbsspiel kann von der ganzen Familie gespielt werden. Anders als andere Computerspiele enthält es keine Zahlen oder Arithmetik. Am Anfang des Spieles geben Sie die Zahl der Spieler ein, und dann fängt es an.

Der Computer wird Ihnen sieben Buchstaben geben:

Wenn er Sie fragt:

V ODER K?

geben Sie V für einen Vokal und K für einen Konsonanten ein.

Wenn Sie Ihre sieben Buchstaben beisammen haben, fordert er Sie auf, daraus ein möglichst langes Wort zu machen, wobei Sie jeden Buchstaben nur einmal verwenden dürfen. Es gibt eine Zeitgrenze, sie wird auf einer Uhr auf dem Bildschirm angezeigt. Erstaunlich ist, wie schnell man diesem Spiel verfallen kann.

Probieren Sie, und Sie werden sehen.



```
10 rem * c-64 vokale und konsonanten *
20 print "{clr}";
30 input "zahl der spieler";np
40 if np<1 then 20
50 dim po(np),c$(7),wo$(20)
60 goto 1000
100 rem * buchstaben waehlen *
110 le$=""
120 print "{down}zug des spielers";pl
130 print "v oder k?"
140 for du=1 to 7
150 get in$
160 if in$<>"v" and in$<>"k" then 150
170 if in$="v" then ch$="aeiou"
180 if in$="k" then ch$="bcdfghjklmnpqrs
tvwxyz"
190 c$=mid$(ch$,rnd(0)*len(ch$)+1,1)
```

```
200 le$=le$+c$
210 print c$;
220 next du
230 return
300 rem * wort eingeben *
310 le=0
320 print "▒zeit in sekunden"
330 for tm=45 to 0 step -1
340 print "▒▒▒▒▒▒▒         ";:if tm<10 th
en print " ";
345 print tm
350 for de=1 to 48
360 get in$
370 if (in$<>chr$(20) or le=0) and (in$<
"a" or in$>"z") then 410
375 if le>6 and in$<>chr$(20) then 410
380 print "▒▒▒▒▒▒▒▒▒▒";tab(le+1);in$
390 if in$=chr$(20) then le=le-1:goto 41
0
400 le=le+1:wo$(le)=in$
410 next de,tm
420 return
500 rem * punkte berechnen *
510 for c=1 to 7
520 c$(c)=mid$(le$,c,1)
530 next
540 for co=1 to le
550 for c=1 to 7
560 if wo$(co)=c$(c) then c$(c)="":goto
580
570 next c:goto 600
580 next co
590 po(pl)=po(pl)+le
600 print"▒▒▒▒▒▒▒▒▒▒▒▒";
610 for p=1 to np
620 print"spieler";p;":";po(p);" punkte"
630 next p
640 return
1000 rem * hauptprogramm *
1010 for pl=1 to np
1020 gosub 100:rem buchstaben waehlen
1030 print"▒bilden sie ein wort aus▒":pr
int tab(6);le$
1040 gosub 300:rem worteingabe
1050 gosub 500:rem punkte berechnen
1060 next pl
1070 goto 1010
1080 end
```

Astrologie

Dieses Programm baut auf einer Untersuchung auf, die vom holländischen Physiker und Astrologen Dr. Ing. van Slooten durchgeführt wurde. Er war Entwicklungsingenieur bei Philips, der seine gesamte Freizeit mit Astrologie zubrachte. Er entwickelte eine Theorie, die besagt, daß die Mondphase zum Geburtszeitpunkt einen sehr wichtigen astrologischen Einfluß auf den persönlichen Charakter habe. Nach einer Untersuchung von Hunderten von Menschen zog er den Schluß, "... daß die Mondphase das Ausmaß bestimmt, zu dem sich das geistige und emotionelle Leben entwickeln wird, insbesondere unter Berücksichtigung des persönlichen sozialen Umfeldes, und darüber hinaus die Rolle, die er oder sie in unserer Gesellschaft spielen wird ..."

Das Diagramm zeigt die Zyklen der Mondphasen. Da es drei Phasen gibt, zunehmend, voll und abnehmend, können drei Typen von Personen festgelegt werden.

Individualisten:

Geboren bei zunehmendem Mond haben sie zunehmende Energie. Sie arbeiten gerne selbständig, haben einen starken Willen, und werden durch körperliche Unbequemlichkeiten nicht entmutigt.

Sozialisten:

Geboren in der Vollmondphase, leben sie gerne in Gemeinschaften. Schlüsselwörter für diese Leute sind "vergleichen", "wählen" und "kombinieren". Kooperation ist ein dominierender Faktor, aber es gibt auch Rivalität und Neid.

Konservative:

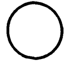
Geboren unter abnehmendem Mond, wissen sie, daß das Licht abnimmt, aber wieder zurückkehren wird. Sie sind sorgfältig und selbstsicher und haben gerne alles unter Kontrolle.

Diese drei Typen können weiter unterteilt werden; man erhält zwölf Kategorien insgesamt.

Pioniere sind Sucher, immer der Masse voraus. Die Stärke ihrer Willenskraft läßt sie manchmal eigensinnig scheinen.



Koordinatoren sind gerne unter Menschen, aber fühlen, daß Sie "gleicher" sind als ihre Mitmenschen. Sie sind romantisch und Familienleben zieht sie sehr an.



Realisten schauen auf die Welt in einer wirklichkeitsnahen Weise. Sie denken gerne, daß alles mit rein physikalischen Begriffen erklärt werden kann.



Spiritualisten sind ziemlich "jenseitige" Charaktere. Philosophisch von der Auffassung her sind sie auf der Suche nach dem Unbekannten.



Künstler verwandeln Materialien in einer kreativen Weise. Sie können Kunstwerken, neuen Produkten oder irgend etwas, was es vorher nicht gab, Gestalt geben.



Apostel sind nicht selbst kreativ, sondern verbreiten die Ideen von anderen.



Helfer nehmen Verantwortlichkeiten für ihre Mitmenschen auf sich, die sonst möglicherweise ignoriert würden. Sie sind hilfsbereit und kümmern sich um andere.



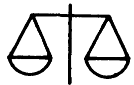
Führer nehmen auch Verantwortung auf sich, aber in diesem Fall für die Leitung von Leben und Taten anderer. Sie schauen vorwärts, wo die meisten Leute es vorziehen würden, rückwärts zu schauen.



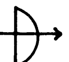
Ritter sind furchtlose Abenteurer, immer auf der Suche nach neuen Dingen. Mit ihrer idealistischen Auffassung haben sie ein starkes Ehrgefühl.



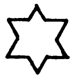
Gebieter führen gerne den Vorsitz bei Taten anderer. Sie urteilen über Dinge nicht voreilig, aber wenn sie eine Entscheidung treffen, erwarten sie, daß sie befolgt wird.



Revolutionäre sind mit der Welt, so wie sie jetzt ist, unzufrieden. Sie neigen dazu, die Dinge negativ zu sehen, und haben das Verlangen, die Welt zu verändern.



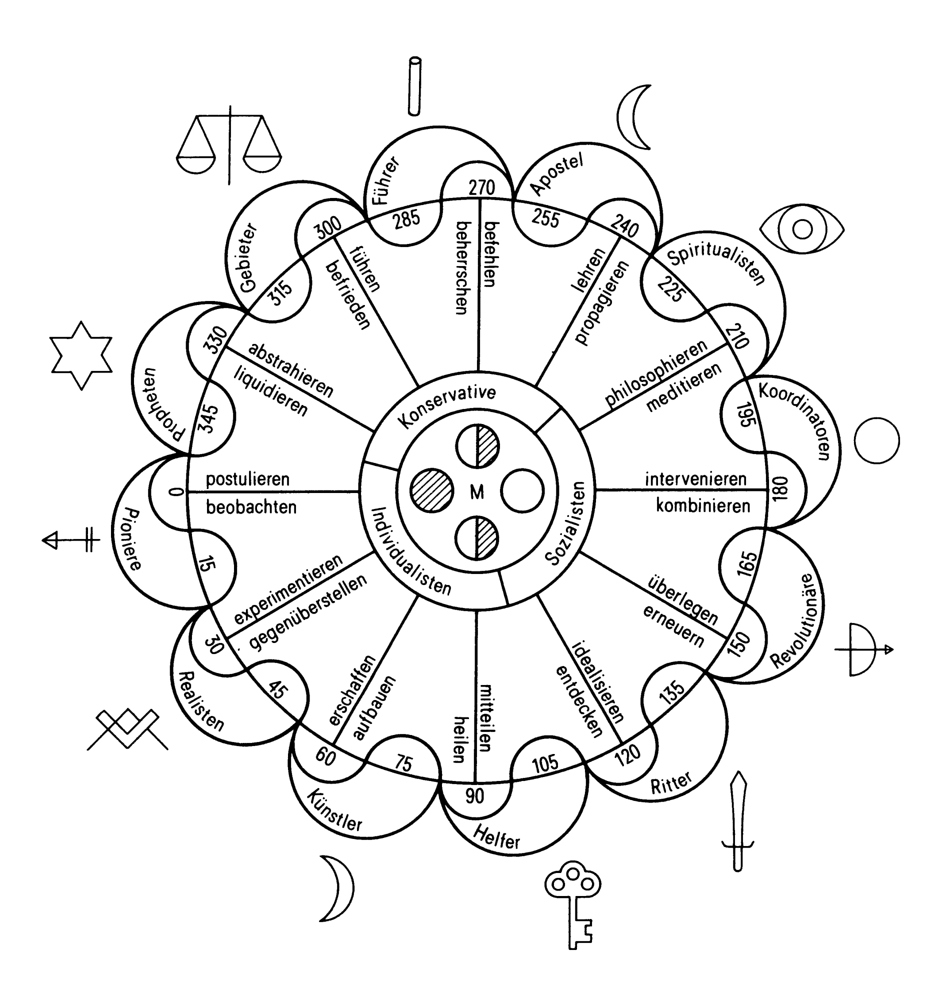
Propheten beschauen die Welt "von oben" und prophezeien die Zukunft. Auf sie wird natürlich nicht immer gehört.



Das Programm sagt Ihnen die Mondphase am Tag Ihrer Geburt. Daraus können Sie die Geheimnisse Ihres Charakters entdecken!

```
10 rem * c-64 mondphasen astrologie *
20 goto 1000
100 rem * datumseingabe *
110 print "{clr}mondphase{down}{down}"
120 input "tag   tt ";dd
130 input "monat mm ";m
140 input "jahr jjjj";y
150 if m<3 then m=m+12:y=y-1
160 return
200 rem * berechnung *
210 t=int(365.25*y)+int(30.6*(m+1))+dd-6
94038
220 t=t/36525
230 la=350.737486+1236*t*360
240 la=la+307*t+6*t/60
250 la=la+51.18*t/3600-5.17*t*t/3600
260 la=la-int(la/360)*360
270 la=int(la+.5)
280 return
300 rem * anzeige des ergebnisses *
310 print "{down}{down}mondphase="la" grad{down}{down}"
320 return
```

```
1000 rem * hauptprogramm *
1010 gosub 100:rem datumseingabe
1020 gosub 200:rem berechnung
1030 gosub 300:rem ergebnis anzeigen
1040 print "druecken sie eine taste"
1050 get a$:if a$="" then 1050
1060 goto 1010
```



Führer
270
Apostel
Gebieter
300
285
befehlen
beherrschen
255
240
führen
befrieden
lehren
propagieren
Spiritualisten
315
225
330
abstrahieren
liquidieren
210
philosophieren
meditieren
Propheten
345
Konservative
Koordinatoren
195
0
postulieren
beobachten
M
intervenieren
kombinieren
180
Pioniere
15
Individualisten
Sozialisten
165
experimentieren
gegenüberstellen
überlegen
erneuern
Revolutionäre
30
150
Realisten
45
erschaffen
aufbauen
mitteilen
heilen
idealisieren
entdecken
135
60
75
105
120
90
Ritter
Künstler
Helfer